Heuristics for the Critical Node Detection Problem in Large Complex Networks

Mahmood Edalatmanesh

Department of Computer Science

Submitted in partial fulfilment
of the requirements for the degree of

Master of Science

Faculty of Mathematics and Science, Brock University
St. Catharines, Ontario

©June, 2013

Abstract

Complex networks have recently attracted a significant amount of research attention due to their ability to model real world phenomena. One important problem often encountered is to limit diffusive processes spread over the network, for example mitigating pandemic disease or computer virus spread. A number of problem formulations have been proposed that aim to solve such problems based on desired network characteristics, such as maintaining the largest network component after node removal. The recently formulated critical node detection problem aims to remove a small subset of vertices from the network such that the residual network has minimum pairwise connectivity. Unfortunately, the problem is NP-hard and also has $\mathcal{O}(|V|^3)$ constraints, making very large scale problems impossible to solve with traditional mathematical programming techniques. Even many approximation algorithm strategies such as dynamic programming, evolutionary algorithms, etc. all are unusable for networks that contain thousands to millions of vertices. A computationally efficient and simple approach is required in such circumstances, but none currently exist. In this thesis, such an algorithm is proposed. The methodology is based on a depth-first search traversal of the network, and a specially designed ranking function that considers information local to each vertex. Due to the variety of network structures, a number of characteristics must be taken into consideration and combined into a single rank that measures the utility of removing each vertex. Since removing a vertex in sequential fashion impacts the network structure, an efficient post-processing algorithm is also proposed to quickly re-rank vertices. Experiments on a range of common complex network models with varying number of vertices are considered, in addition to real world networks. The proposed algorithm, *DFSH*, is shown to be highly competitive and often outperforms existing strategies such as Google PageRank for minimizing pairwise connectivity.

Acknowledgements

I am very grateful of people who helped me through all difficulties of this work by their different contributions. Firstly I would like to thank my supervisors Dr. Ombuki and Dr. Ventresca for all their help and guidance. I would also like to thank my family and my friends who supported me morally and technically through my work. Last but not least, I would like to thank Cale Fairchild who made my life a lot easier by his technical helps.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The main objective of this thesis is to develop efficient heuristics for the critical node detection problem (CNDP) with specific application to very large networks. A complex network is a graph where the connections between nodes have an inherent real-world meaning. Examples of natural or artificial systems that can be represented as complex networks include metabolic networks [44], the World Wide Web [8], and social networks [62, 75]). In many cases real-world networks are often large [59] (i.e., with thousands to millions of nodes), and thus their large size needs to be taken into consideration.

Removing nodes randomly from a graph and studying effects of such removals on connectivity of graph has been studied extensively for regular graphs [16]. However, many critical node detection problems have been defined in the literature in order to find some nodes of the input graph such that their removal lead to a desired objective [2, 23, 24, 46, 58, 69, 77]. Finding critical nodes of real networks has attracted much attention in recent years due to their crucial application in the real world. Some of the important applications of finding critical nodes are to prevent the spread of an infection in society, a computer virus in the Internet, or a rumour in a social network. The criticality of nodes must be defined in order to find the nodes of a network that their removal result in minimizing the spread of an infection for those applications. Researchers have studied the effect of removing critical nodes using different strategies on various real networks [2, 23, 24, 46, 58, 69, 77]. Different criteria were used to find the most critical nodes of the input network for different objectives (e.g., size of the largest component [69], average shortest path value [23, 24], and the diameter [2]).

Although variants of the critical node detection problem have been studied before, this thesis focuses on a recently proposed CNDP [4]. Borgatti [17] proposed a new definition of critical node based on pairwise connectivity after removing a certain number of nodes from a network. This problem was later formally defined by Aruleslvan et al. [4] and it was called the critical node detection problem (CNDP). The CNDP aims to find a subset $L \subseteq V$ of critical nodes where $|L| \leq k$ (the number k is given by user) to be removed from a given graph. The aim is to find the k critical nodes whose removal results

in minimum pairwise connectivity.

The CNDP has many applications in the real world. The applications involve situations where the aim is to either protect the connectivity of nodes in a network by securing the most critical nodes or attacking the most critical nodes in order to have minimum connections between all pairs of nodes in the network. Some important applications of the CNDP are presented below.

In a supply chain network, the connections between pairs of nodes is minimized after removing the most critical nodes from the network. For example, a military supply chain network [80] contains battalions and support battalions as nodes and the connections between them as links. By attacking the most critical nodes in this network, the connectivity between supply and demand nodes will be minimized. Therefore, the solution of the CNDP is important in military tactical attacks during wars.

By using the gathered intelligence from a covert network, the terrorist network can be represented as a graph where terrorists are depicted as nodes and the social interactions between them represent links. We can minimize the communications between terrorists by attacking the most critical individuals in the networks [4].

People and contacts between them in a real society are represented as nodes and links of a graph in order to study the effect of epidemics in real social network [57, 68]. In order to prevent the spread of infectious diseases in real social networks, different strategies were presented for targeted vaccinations since random mass vaccinations are expensive [57, 68]. However, the optimal vaccination strategy is to find the critical nodes and vaccinate them to minimize the pairwise connectivity between people in a society [4], assuming that higher pairwise connectivity cause faster outbreak.

Telecommunication networks such as the Internet, telephone networks, and computer networks can be represented as graphs, where each node is a terminal and links show the communications between terminals. In telecommunication networks like the Internet, the information spreads between all nodes that there is a path between them (they are pairwise connected). Therefore, the CNDP is important for these networks to find the critical nodes that their removal result in maximum communication breakdowns [4]. Furthermore, in order to prevent the spread of viruses over telecommunication networks, more protection must be provided for the critical nodes [5].

As stated above, the CNDP has many applications in the real world. Since the CNDP is a NP-complete problem [4], heuristics are necessary in order to approximate the problem within practical time. Different heuristics have been proposed for the CNDP such as simulated annealing [72], population based incremental learning [72], genetic algorithm [14], and a combinatorial heuristic with local search [4]. In all previous works on the CNDP, the heuristics were evaluated on small networks (of size at most 5000) even though most real-world complex networks are often large (with hundreds of thousands to millions of nodes) [59]. Hence, the main motivations for this thesis are as follows:

- As indicated above, there are many applications where finding critical nodes in a real-world network is crucial.
- The CNDP is an NP-complete problem and efficient heuristics need to be designed to find good solutions within reasonable time.
- The previously designed heuristics did not focus on approximately solving the CNDP for large networks (i.e., with hundreds thousands to millions of nodes), where the size of many real-world networks is in this range [59].

1.1 Goals

Since many applications of the CNDP are in large complex networks, there is a need to have an approximate solution to the problem feasible in real large networks (e.g., a phone call network with 53 million nodes [1]). The main goal of this thesis is to design and develop a fast heuristic for the CNDP that is also competitive to previous methods in the literature in respect to computational time and quality of solution [23, 24, 35, 57].

The real-world complex networks may have different topological features than each other, which may affect the performance of an approach. Therefore, the aim is to design a heuristic that is flexible for different network topologies in order to maintain the quality of the solution of the heuristic.

1.2 Challenges and Contributions

Since the CNDP has only been recently (2009) formally defined, many of its properties are still not well understood, e.g., it was only defined for undirected unweighted networks. Moreover, given that the CNDP is a NP-complete problem, designing practical solutions especially for very large networks is considered a challenging task. One of the main challenges in this thesis is to design a ranking function that ranks the nodes of the input graph based on the objective of the CNDP. Due to the constraints on the time complexity, many useful graph properties such as closeness centrality and betweenness centrality measures are not suitable to be used in the ranking procedure because of their computational time. It takes hours to compute algorithms of complexity within $\mathcal{O}(|V|^2)$ for large networks (with millions of nodes).

In summary, the main contributions of this thesis are:

1. Developing a fast heuristic of complexity lower than $\mathcal{O}(|V|^2)$ for the CNDP feasible in large complex networks (i.e., with hundreds of thousands to millions of nodes).
2. Designing a ranking function with the flexibility for application to different network topologies.

3. Developing a post-processing procedure to boost the performance of the heuristic proposed in this thesis based on the objective of the CNDP.

4. Compare the results of presented heuristics to known centrality-based measures and some previous heuristics [72] in order to evaluate the performance of the designed heuristics.

Different benchmark suites are utilized and proposed in this thesis listed as: small networks of sizes ranging from 500 to 2000, small to larger size networks of sizes ranging from 100 to 25,000, and large real-world networks of size ranging from thousands to millions of nodes. The comparisons show that the heuristics presented in this thesis outperform other approaches in most benchmark suites and real-world networks.

1.3 Thesis Structure

The rest of this thesis is organized as follows. Background information on graphs, the definition of the CNDP, properties of complex networks, centrality-based approaches, and the literature review on the CNDP are given in Chapter 2. The methodology, ranking function, and post-processing procedure are described in Chapter 3. The information about the benchmark models used in this thesis and the results of comparisons between the performance of the proposed heuristic and other approaches on benchmark networks are given in Chapter 4. Experimental results on real-world networks are presented in Chapter 5, with conclusions and future work given in Chapter 6.

Chapter 2

Background

This chapter presents a summary of background information on complex networks and their properties that are relevant to this thesis. Furthermore, the graph properties related to the objective of the CNDP are also introduced. In order to assess the performance of the proposed heuristics, they are compared to different centrality-based approaches, which are described here as well. In addition, a literature review of previous and related works on the CNDP and other variants of “critical node" definition are given.

2.1 Graphs

A graph $G = (V, E)$ is a pair (V, E) such that V is the set of vertices and E is the set of edges, where each edge is an unordered pair of vertices from set V. The number of vertices and edges of a graph are calculated by the cardinality of sets V and E, respectively. Many real-world situations can be represented as a graph depicting objects as nodes and the relationship between any two objects as edges. If the cost of having an edge between any pair vertices in a graph is not the same, it is called a weighted graph.

2.1.1 Special Graphs

In order to highlight the difference between the structure of complex networks and other graph structures, some well-known graphs are described here.

Complete Graphs

A complete graph G with $|V| = n$ nodes contains $|E| = \frac{n(n-1)}{2}$ edges, which means that each node is connected to every other node. A complete graph with 5 nodes is shown in Figure 2.1(a).

Regular Graphs

A regular graph is one where the number of neighbours of each node is the same. That is, each node in a c-regular graph is connected to c other nodes. Figure 2.1(b) shows a 3-regular graph with 6 nodes.

Trees

A tree is a connected graph with $|V| = n$ vertices and $|E| = n-1$ edges, where a graph is called connected if there is a path between each pair of nodes. No cycle exists in a tree and the deletion of any node $u \in V$, except for nodes of degree 1, increases the number of components in the induced subgraph $G(V \setminus \{u\})$. Figure 2.1(c) shows an example of a tree with 5 nodes.

Star Graphs

A star graph is a tree that has $|V| = n$ nodes and $|E| = n-1$ edges. In star graphs, one node has degree $n-1$ and all others have degree 1. A star graph with 5 nodes is shown in Figure 2.1(d).

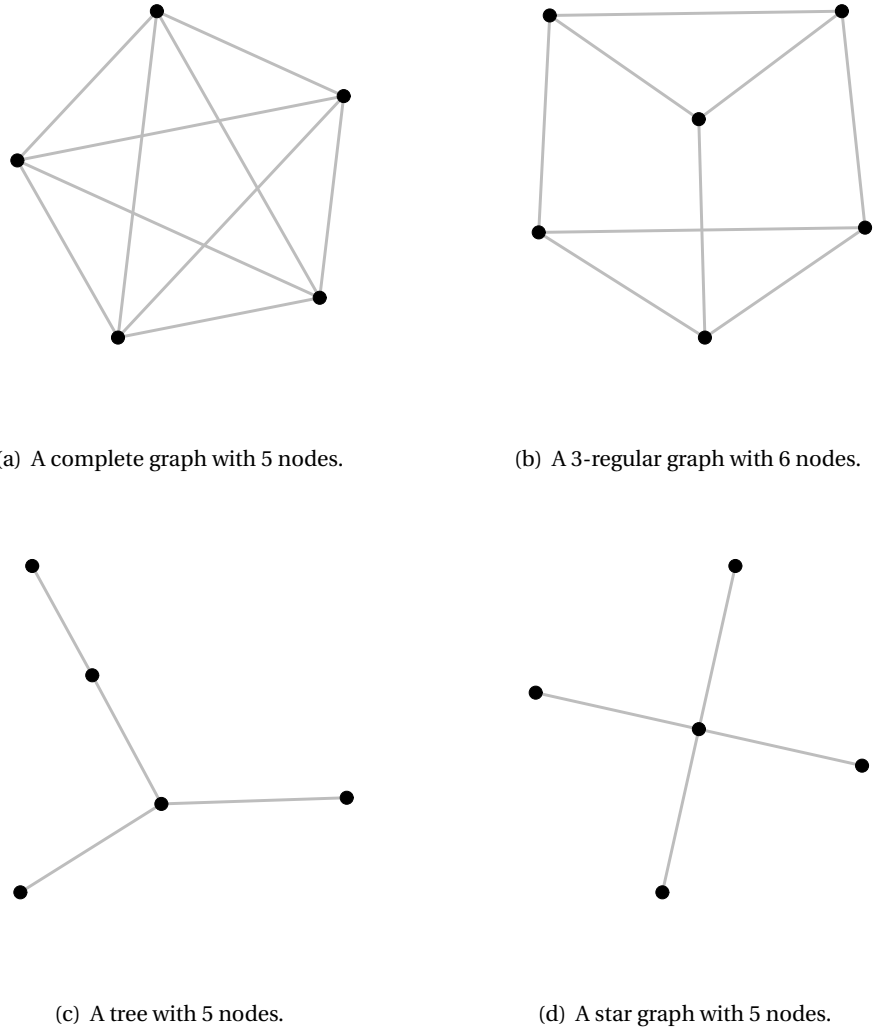
2.1.2 Random Graphs

A random graph is a graph generated by a random process. Erdös and Renyi [28] proposed a random network model that generates random networks with n nodes and adds an edge between any two nodes with probability p. A sample Erdös-Renyi (ER) network with 35 nodes and $p = 0.07$ is shown in Figure 2.2. As can be seen, the ER model may produce a disconnected graph since the probability of having an edge between any two nodes is the same. Therefore, there is no guarantee that the graph is connected or even all nodes have degree higher than 0. Different random graph models were used in this thesis to produce benchmarks for the purpose of evaluating the performance of proposed heuristics, any needed number of networks of any size can be easily generated by network models.

2.1.3 Complex Networks

A complex network is a graph that has topological features that are not necessarily represented in simple networks such as regular graphs or random graphs, while the features of complex networks such as the small world property, scale-free property, and community structure (See Sub-Section 2.3) are often observed in real-world networks. Moreover, the connections between the vertices of complex networks have an inherent meaning. Complex networks have actively been studied in different fields (biology [7, 14, 44], chemistry [3, 40, 73], telecommunications [25, 67], etc.) due to the need of

understanding and analysing vast number of natural and artificial networks. In real complex networks [13], the nodes represent real objects such as people, and the links connecting them have some meaning in real world, e.g., each link in a social network of acquaintance represents the friendship between two people. The term *complex* refers to the non-trivial topological structure of this kind of network that do not occur in regular networks or random networks. Figure 2.3 shows a complex network of acquaintances between 35 members of a karate club [79], where each node represents a member of the karate club and two members are connected to each other via a link if they are friends.



(a) A complete graph with 5 nodes.

(b) A 3-regular graph with 6 nodes.

(c) A tree with 5 nodes.

(d) A star graph with 5 nodes.

Figure 2.1: Samples of different kinds of graphs.

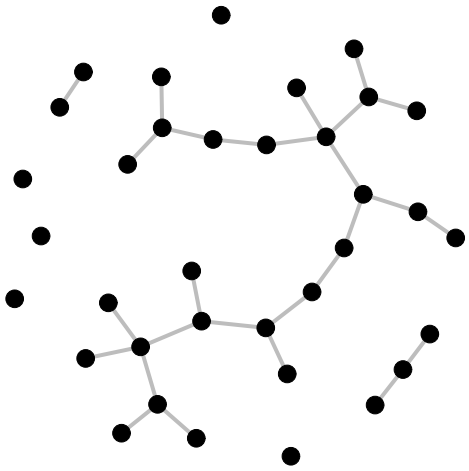


Figure 2.2: An Erdös-Renyi network with 35 nodes and 26 edges with $p = 0.07$.

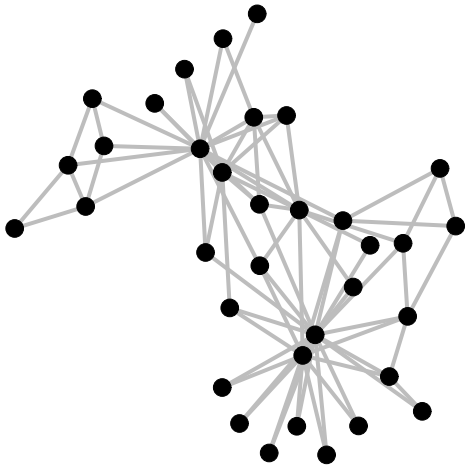


Figure 2.3: The Zachary karate club complex network with 35 nodes and 78 edges [79].

2.2 The Critical Node Detection Problem

The critical node detection problem was formally defined by Aruleslvan et al. [4] in 2009. This definition was derived from the work done by Borgatti who studied critical node detection based on maximum network disconnectivity [17].

The input graph $G = (V, E)$ is assumed to be unweighted and undirected. A number $k > 0$ is given as input, which is the maximum number of nodes that can be removed. The output is a subset $L \subseteq V$, where $|L| \leq k$, whose deletion from the graph minimizes pairwise connectivity among the nodes in the induced subgraph $G(V \setminus L)$. Two nodes in a graph are pairwise connected if there is a path between them. Mathematically, the objective of the CNDP is to determine

$$L = \underset{L \subseteq V}{\operatorname{argmin}} \sum_{i,j \in (V \setminus L)} u_{ij}(G(V \setminus L)) : |L| \leq k, \tag{2.1}$$

where

$$u_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are in the same component of } G(V \setminus L), \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

As given in Eq. (2.2), the measure of pairwise connectivity of the graph is calculated by u_{ij} which is a binary value, and it is equal to 1 if there exists a path between nodes i and j. The objective function stated in Eq. (2.1) [4] can also be revised as:

$$\sum_{m \in M} \frac{\epsilon_m(\epsilon_m - 1)}{2}, \tag{2.3}$$

where M is the set of all connected components in $G(V \setminus L)$, and ϵ_m is the size of component m. The size of all connected components in the residual graph can be calculated by a depth first search or breath first search with complexity $\mathcal{O}(|V| + |E|)$ [21]. Hence, the objective value of a given solution can be calculated in $\mathcal{O}(|V| + |E|)$.

In real-world complex networks, the number of edges is in the order of the number of nodes $\big(|E| = \mathcal{O}(|V|)\big)$, and by even deletion of $k = 50\%$ of nodes, the remainder graph may have many isolated nodes. The applications of the CNDP in real-world networks are interested in small k numbers ($k \ll |V|$) since the size of these networks can be dramatically reduced by deletion of small number of nodes (i.e., $k \leq 50\%$). Moreover, for removing nodes from real-world networks such as the Internet, social networks, or terrorist networks we need to spend a considerable amount resources, and therefore the k-value is usually small in applications of the CNDP.

As an example, an anti-terrorist government needs to spend a considerable amount of resources in order to remove a member of a terrorist network, and therefore it is logical to spend the resources on removing the most critical members.

The optimal solution of the CNDP can be determined by using an an integer programming (IP) formulation of the problem [4]. Since the IP solution is an exponential time algorithm, the optimal solution of the CNDP can only be determined for small networks (with less than 200 nodes), the program will either run out of memory or time in bigger network instances. The IP formulation of the CNDP is described in below.

2.2.1 Integer Programming Formulation

The binary value v_i is defined as:

$$v_i = \begin{cases} 1 & \text{if node } i \text{ is deleted in the optimal solution,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

So the critical node detection problem can be defined as:

$$Minimize \quad \sum_{i,j \in V} u_{ij} \tag{2.5}$$

$$subject\,to \quad u_{ij} + v_i + v_j \geq 1, \forall (i,j) \in E, \tag{2.6}$$

$$u_{ij} + u_{jw} - u_{kw} \leq 1, \forall i,j,w \in V, \tag{2.7}$$

$$u_{ij} - u_{jw} + u_{wi} \leq 1, \forall i,j,w \in V, \tag{2.8}$$

$$-u_{ij} + u_{jw} + u_{wi} \leq 1, \forall i,j,w \in V, \tag{2.9}$$

$$\sum_{i \in V} v_i \leq k, \tag{2.10}$$

$$u_{ij} \in \{0,1\}, \forall (i,j) \in V, \tag{2.11}$$

$$v_i \in \{0,1\}, \forall (i) \in V, \tag{2.12}$$

where u_{ij} has the same definition as given in Eq. (2.2).

The objective of this model is to find a set L of k nodes whose removal cause minimum pairwise connectivity in the induced subgraph $G(V \setminus L)$. Constraint (2.6) means that if nodes i and j are in different components and there is an edge between them, then one of them should be deleted. Constraints (2.7), (2.8), and (2.9) altogether indicate that if nodes i and j are in the same component and also nodes j and w are in the same component, then nodes i and w should also be in the same component. Constraint (2.10) guarantees that the number of nodes to be deleted is at most k. At last, equations (2.11) and (2.12) determine the domain for the decision variables.

The IP model was used to compare the optimal solution with the heuristics defined in different papers [4, 6, 14] since the size of sample networks was small (at most 150). It was not possible to calculate the results of the IP model for networks of size larger than 150 with the available resources in this thesis. The number of constraints is cubic in number of vertices and linear in number of edges.

2.3 Complex Networks Properties

In recent years, the study of large sized networks has generated much interest due to the fact that large complex networks such as the Internet or social networks surround us and the number of applications on problems similar to the CNDP abound. The size of some interesting complex networks such as the Internet now exceeds millions of nodes, and solving combinatorial optimization problems similar to the CNDP by exact algorithms is not applicable for these networks. Therefore, heuristics need to be developed for those problems. Understanding the topology of complex networks may be an asset to design fast heuristics for them or to design proper network models that their topology is similar to what observed on many real-world networks.

Two of the topological properties of complex networks that are observed in many real-world networks [74] are the small-world property and the scale-free degree distribution property. These two properties are used in different network models to produce network samples with properties similar to real-world complex networks. Another attribute of complex networks that is of particular interest to the CDNP is called community structure and it is also explained in this section, in real-world networks the most critical nodes are usually the ones connecting communities to each other.

2.3.1 Small-world Property

In many real-world networks, there is a relatively short path between any two nodes (examples given at [74]). This topological feature is known as the small-world property [56], and it is characterized by an average shortest path length D:

$$D = \frac{1}{|V|(|V|-1)} \sum_{i,j \in V, i \neq j} d_{ij}, \tag{2.13}$$

where d_{ij} is the length of shortest path between any two nodes i and j in the network. The average shortest path length D depends at most logarithmically to the network size in small-world networks [75]. The formula given in Eq. (2.13) is not defined when the network is not connected since the d_{ij} for two nodes belonging to two different components is infinity. As stated in [13], a possible solution for this problem is to use an alternative equation:

$$D = \frac{1}{|V|(|V|-1)} \sum_{i,j \in V, i \neq j} \frac{1}{d_{ij}}, \tag{2.14}$$

where $\frac{1}{d_{ij}}$ for any two nodes i and j placed in different components is equal to zero. An electronic circuit [29] with 329 nodes is shown in Figure 2.4 as an example of small-world networks, where the value of average shortest path is 3.17, which is close to $\log(329) \approx 2.51$.

A measure to calculate the tenancy of nodes to cluster together is called the clustering coefficient [59], and it is formulated as:

$$C = \frac{number\ of\ triangles \times 3}{number\ of\ connected\ triples\ of\ nodes}, \tag{2.15}$$

where a triple of nodes contains three connected nodes that contains either two or three edges, and a triangle is a triple with three undirected edges.

As stated in [75], complex networks with the small-world property also have a high clustering coefficient. The clustering coefficient for the electronic circuit in Figure 2.4 is 0.34.

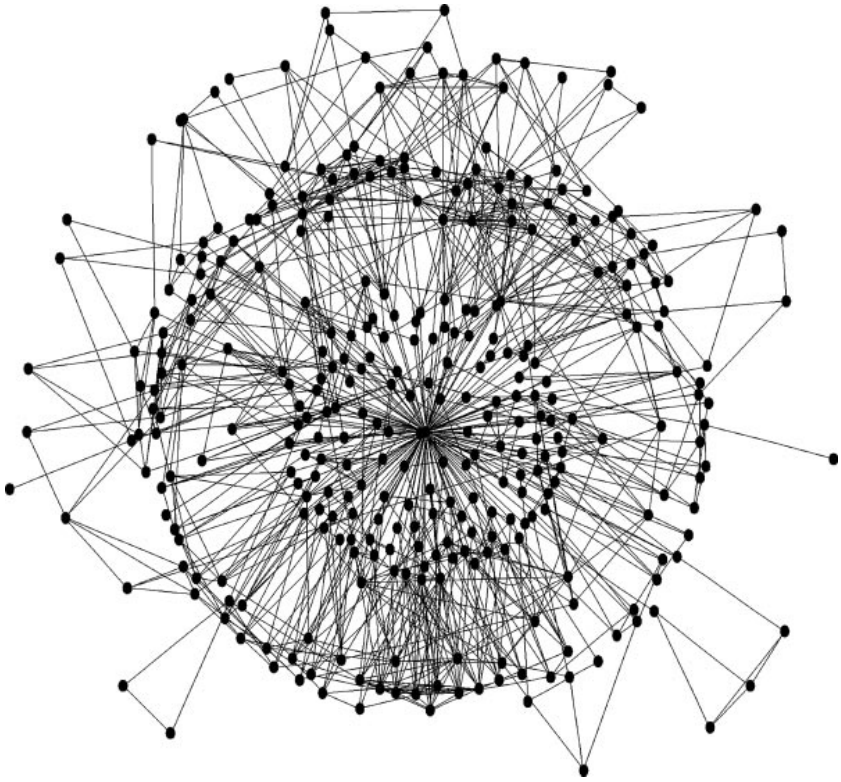
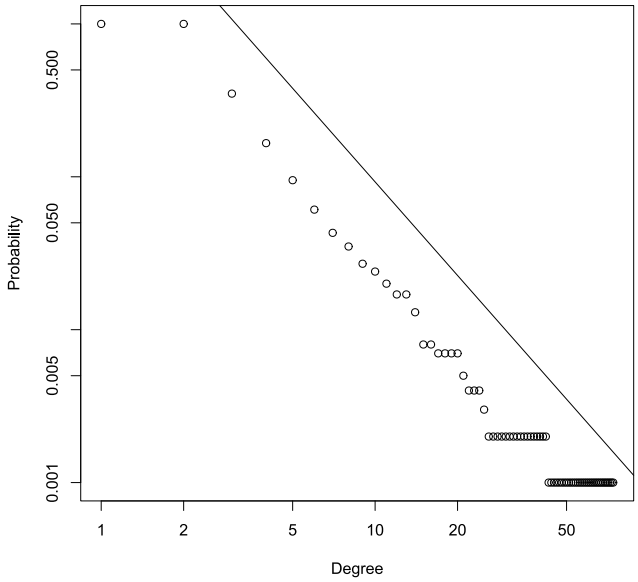


Figure 2.4: A connectivity graph for an electronic circuit with 329 nodes.

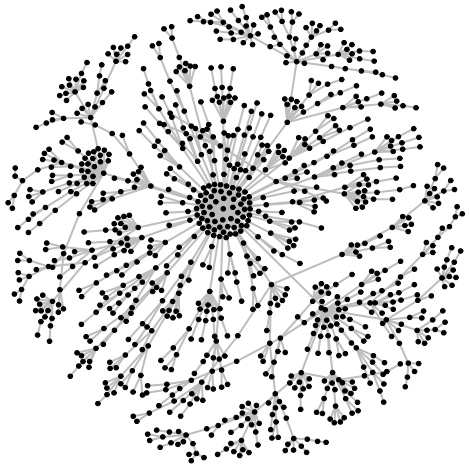
2.3.2 Scale-free Networks

The degree distribution of some complex networks is different than regular or random networks. In these networks the degree distribution $P(x)$ varies as a power of node degree x that follows a power-law degree distribution, $P(x)$ is the fraction of nodes that have degree x to the total number of nodes in graph. That is, the degree distribution of these networks can be formulated as $P(x) \sim Ax^{-\alpha}$ where A is a constant and α is the power-law exponent. The power-law exponent can be any number, but it is usually in the range $2 < \alpha < 3$ based on an empirical study by [13] where they also explained that networks with an exponent in this range have a power law degree distribution. It is important to note that real-world networks may not have the power-law for all degrees [13] and the power-law can only be observed in the tail of the degree distribution [1, 9]. The power-law appears as a straight line with slope $-\alpha$ in the log-log plot of the degree distribution. As an example, the degree distribution of a sample network generated by Barabasi-Albert model [9] is plotted in Figure 2.5(a).

These networks are called scale-free because power-law degree distributions have the same functional form at all scales, which means that as the network grows the log-log degree distribution remains a straight line. In scale-free networks many nodes of small degree exist while only a few nodes of highest degree (hubs) can be observed. For example, Figure 2.5(b) shows a scale-free Barabasi-Albert network containing many nodes of small degrees and only a few nodes of highest degree. Since some different classes of real-world networks (e.g., telecommunications, biological, and social networks [74]) have the power-law distribution, studies were done on these networks in the literature [9, 20]. As a result of these studies, Barabasi et al. [9] proposed a network model that produces networks with power-law degree distribution based on using two mechanisms named as network growth and preferential attachment (this model is described in Chapter 4). Also, Cohen et al. [20] investigated the resilience of scale-free networks against random node failures and the results indicated that scale-free networks are robust (in terms of network connectivity) against removing nodes randomly. This property of scale-free networks follows from the fact that most of the nodes have low degree and a few nodes exist with high degrees. Therefore, if the probability of removing each node is the same as others, the chance of removing the nodes of high degree is low.



(a) The log-log degree distribution of a Barabasi-Albert network with 1000 nodes.



(b) A Barabasi-Albert network with 1000 nodes.

Figure 2.5: A Barabasi-Albert network and a log-log degree distribution.

2.3.3 Community Structure

Real-world networks have inherent meaning that makes them different from regular or random networks. One of the outstanding differences between complex networks and other networks is that the distribution of edges is globally and locally inhomogeneous [32]. This leads to different groups of nodes where each group has many links between each other and a few links to the rest of the graph [32]. These groups are called communities, clusters or modules. A sketch of a network with communities is given in Figure 2.6, where the nodes connecting each community to the rest of the graph are depicted in colors different than black for each community. Due to the high cost of removing k critical nodes from large complex real-world networks (i.e., with hundreds of thousands to millions of nodes), it is important to take advantage of all useful properties of complex networks to design proper heuristics for the CNDP and select the most critical nodes; a useful property is the community structure. As Fortunato reported [32], most of the community detection algorithms are at least of order $\mathcal{O}(|V|^2)$, which is considered expensive for large complex networks. Moreover, there is no guarantee that information of calculated communities can immediately result the k most critical nodes whose deletion minimizes the pairwise connectivity in the residual graph. Hence, ranking the nodes by only using community detection algorithms is not enough for the CNDP.

Many different techniques have been proposed and designed for finding communities in networks. Recently, Fortunato [32] made a comprehensive study on various community detection methods where a comparison was done from different aspects, such as time complexity or the ability of methods to detect different kinds of communities (hierarchical, overlapping, etc.). Fortunato [32] stated that researchers in the field of community detection need to agree on a unique definition for communities. Due to the differences between opinions, various scientists have defined the communities based on their own point of view and designed their algorithms regarding to their definition of communities [32]. Therefore, it is hard to compare the efficiency of different algorithms in detecting communities. Fortunato [32] also indicated that having a reliable benchmark graph for testing the algorithms is dependent on the definition of clusters and partitions. Therefore, at this time, it is not easy to determine the most efficient algorithm for detecting communities, and the exact community detection algorithms were not used in the heuristics proposed in this thesis.

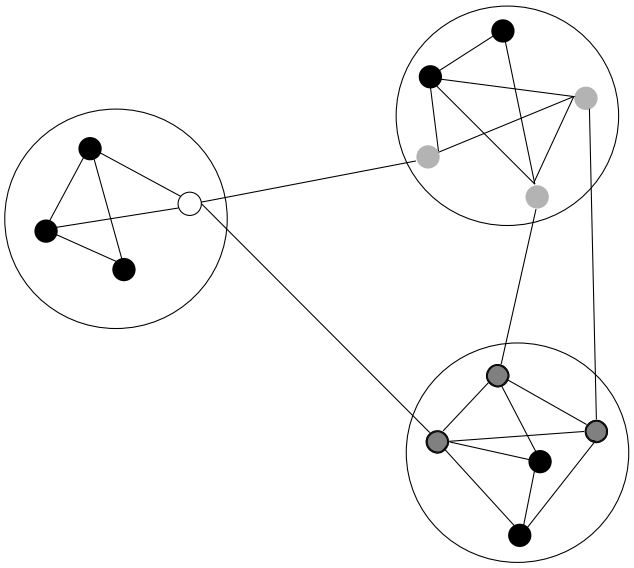


Figure 2.6: A graph with three communities.

2.4 Graph Properties

A variety of graph properties were utilized and the following subsections provide a summary of those relevant to this thesis. It was aimed to find the most suitable properties that help to indicate the critical nodes of a graph in the context of the CNDP.

2.4.1 Cut vertices, Bridges, and Biconnected-components

A node v of a graph G is a cut vertex if removing node v and its incident edges increases the number of disconnected components in G. A bridge e is an edge whose removal disconnects G. A biconnected component is a subgraph of G whose any two edges lie in a cycle. In other words, no cut vertex or bridge exists in a biconnected component [21]. The biconnected components, cut vertices, and bridges of a graph G can be determined during a pass on all nodes of G by depth first search (DFS) algorithm with computational complexity $\mathcal{O}(|V|+|E|)$ [21].

The following lemma was proven in [4], which indicates that increasing the number of disconnected components results in a better objective value of the CNDP.

Lemma 2.4.1. *Let M_1 and M_2 be two sets of partitions obtained by deleting L_1 and L_2 sets of nodes, respectively, from graph $G = (V, E)$, where $|L_1| = |L_2| = k$. Let T_1 and T_2 be the*

number of components in M_1 and M_2, respectively, and $T_1 \geq T_2$. If all partitions in M_1 have the same size, then we obtain a better objective function value by deleting the set L_1.

It is important to identify the cut vertices and bridges of the network since the number of disconnected components or isolated nodes (nodes of degree 0) increases after removing cut vertices from the network. However, if cut vertices belong to a small component, then the overall impact may not be as good as removing non-cut vertices from a very large component. The cut vertices and bridges are the connections between biconnected components, and therefore prioritizing removal of cut vertices is likely beneficial. Moreover, a cut vertex with more incident bridges is more important than other cut vertices with lower number of incident bridges, which indicates the necessity of using the information about bridges of graph G to calculate the k critical nodes in this thesis. Unfortunately, these attributes are not enough to solve the problem due to the fact that in many cases the k critical nodes of a graph are not all cut vertices. As an example, the optimal solution of the CNDP for a network of size 100 with $k = 20$ is shown in Figure 2.7, where the white nodes represent the selected nodes. The double circled white nodes are the selected non-cut vertices.

2.4.2 Vertex Similarity

Vertex similarity measures the similarity of two target vertices u and v in a graph. Similarity measures are based on common neighbours between nodes $u, v \in V$. Zhou et al. [81] investigated nine different similarity measures on real-world complex networks. The results of the experiments showed that two of the most accurate measures are the Jaccard similarity coefficient [43] and the Sorensen-Dice similarity coefficient [70]. A neighbourhood of a vertex must be defined in order to introduce these two similarities. The neighbourhood $\Gamma(v)$ of a vertex $v \in V$ is the set of all nodes in the network that are connected to node v via an edge:

$$\Gamma(v) = \{w \in V \mid (v, w) \in E\}. \tag{2.16}$$

The degree $deg(v)$ of node v is the number of neighbours of node v:

$$deg(v) = |\Gamma(v)|. \tag{2.17}$$

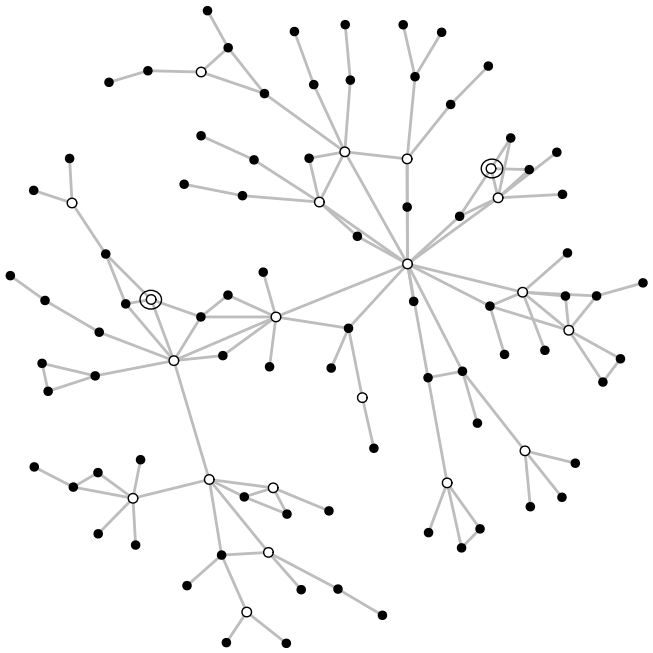


Figure 2.7: A network sample with 100 nodes, where the white nodes represent the optimal solution when $k = 20$.

The Jaccard similarity [43] of two vertices v and u is the number of common neighbours between them divided by the total number of neighbours they have (common and uncommon altogether) which is defined as:

$$V_J(v,u) = \frac{|\Gamma(v) \cap \Gamma(u)|}{|\Gamma(v) \cup \Gamma(u)|}. \tag{2.18}$$

The Sorensen-Dice similarity [70] of two vertices v and u is twice the number of their common neighbours divided by the sum of the degrees of the two vertices:

$$V_S(v,u) = \frac{2|\Gamma(v) \cap \Gamma(u)|}{deg(v) + deg(u)}. \tag{2.19}$$

Based on the results of experiments in [81], there is no difference in the accuracies of the two similarity measures mentioned above, and the Sorensen-Dice similarity was used to compute the vertex similarity in this thesis. The value of $V_S(v,u)$ is in the range $[0,1]$, where higher values means that the two nodes are more similar to each other than other pairs of nodes. An edge is called a local bridge if its endpoints have no neigh-

bour in common; in this case $V_S(v,u)$ is equal to zero. Onnela et al. [62] compared the effect of removing local bridges against randomly picked edges over a large cellphone network with four million nodes, and the results showed that removing local bridges is more effective than randomness in order to break the network into more disconnected components [62]. As given in Lemma 2.4.1, increasing the number of disconnected components in the residual graph results in a better objective value for the CNDP.

The vertex similarity measures can help to measure how much each node is locally similar to its neighbours, and this can lead to determine if a node has strong connections to its neighbours or if it is part of a local bridge to another community. However, local bridges are not necessarily only between communities, e.g., the vertex similarity between two endpoints of any edge of a ring with $|V| > 3$ is zero, while no local bridge exists in a ring. Regarding the objective of the CNDP, removing the vertices with lower similarity to their neighbours is more likely to help to decrease the pairwise connectivity in the residual graph since higher similarity values for a node means that it has strong connections to its neighbours.

2.5 Centrality-based Approaches

2.5.1 Degree Centrality

The degree centrality provides information about the degree of each node (Eq. (2.17)) based on the idea that nodes with higher degrees are more important in the network. The degree centrality of any node $v \in V$ is formulated as [45]:

$$C_D(v) = \frac{deg(v)}{(|V|-1)}. \tag{2.20}$$

The time complexity for calculating the degree centrality of all nodes in a graph is $\mathcal{O}(|V|^2)$ in dense graphs and $\mathcal{O}(|E|)$ in sparse graphs [45].

2.5.2 Closeness Centrality

The closeness centrality measures the importance of each node in spreading information to other nodes based on the total shortest path length between that node and all other nodes. Nodes in the center of the graph have the lowest total shortest path, and therefore their closeness value is highest. The closeness centrality of node $v \in V$ is defined as [45]:

$$C_C(v) = \frac{|V|-1}{\sum\limits_{i \in V, i \neq v} d_{vi}}, \tag{2.21}$$

where d_{vi} is the length of the shortest path from node v to node i. The time complexity for calculating shortest paths based on Fredman et al. [34], implemented the Dijkestra's algorithm based on a min-priority queue (e.g., a Fibonacci heap), has worst case complexity $\mathcal{O}(|E|+|V|\log|V|)$. Since all pairs of shortest paths need to be calculated for the closeness centrality of one node, the time complexity of the closeness centrality is $\mathcal{O}(|V|\,|E|+|V|^2\log|V|)$.

2.5.3 Betweenness Centrality

Betweenness centrality measures the number of times that a node is in the shortest path between any two other nodes in a graph. Therefore, if two communities X and Y have only one way to communicate to each other (via a bridge), the endpoints of the bridge between them will have a higher betweenness centrality than other nodes of the communities. The betweenness centrality of node $v \in V$ is formulated as [45]:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\rho_{st}(v)/\rho_{st}}{(|V|-1)(|V|-2)}, \tag{2.22}$$

where ρ_{st} is the number of shortest paths from node s to node t, and $\rho_{st}(v)$ is the number of shortest paths from node s to node t that pass through node v. The time complexity for calculating the betweenness centrality is $\mathcal{O}(|V|^3)$ [45], although Brandes [18] proposed a faster algorithm for large sparse networks that runs in $\mathcal{O}(|V|\,|E|+|V|^2\log|V|)$ and $\mathcal{O}(|V|\,|E|)$ on weighted and unweighted networks, respectively.

2.5.4 PageRank

An approach was proposed to rank each node of a graph based on its degree and the rank of its neighbours, which is called the PageRank [19]. The idea behind the PageRank centrality measure is that a node has high rank if the sum of the ranks of nodes connected to it by inedges (in directed networks) is high [63]. In undirected networks, the rank of each node is calculated based on the sum of the ranks of its neighbours. The formula that calculates the PageRank of node i in undirected graphs is as follows:

$$PR(i) = \frac{1-b}{|V|} + \sum_{j \in \Gamma(i)} \frac{PR(j)}{deg(j)}, \tag{2.23}$$

where $PR(i)$ is the PageRank of node i, and b is the damping factor, which is the probability that surfing the network would continue (this number is suggested by Brin and Page [19] to be 0.85). The nodes with higher ranks are considered more important since they have either many edges or nodes with high ranks are linked to them [63].

2.5.5 Uses of Centrality Based Measures

For about the last half a century, many measures have been proposed for finding the most critical nodes in a network whose deletion would damage the network, e.g., decrease the size of the largest component, increase the average shortest path length, etc. Crucitti et al. [24] studied the vulnerability of complex networks based on the degree of the nodes and the betweenness centrality. They found that networks with power-law degree distribution (scale-free property) are vulnerable to deletion of central nodes. Freeman et al. [35] used closeness centrality to measure the goodness of central nodes on different network structures such as wheel, circle, or chain, and compared it with distance-based centrality measures such as betweenness centrality. They showed that removing the nodes based on distance-based centralities is more effective than closeness in order to break the network into more disconnected components [35]. Brin et al. [19] proposed PageRank to find the most important webpages in Google's search engine, which was faster than previous engines and more accurate.

It should be noted that three centrality measures (betweenness centrality, closeness centrality, and degree centrality) [24, 35] and PageRank [19] were used in this thesis to evaluate the performance of the proposed heuristics. Since distance-based centrality measures cannot be computed in less than $\mathcal{O}(|V|^2)$ they were not used in heuristics proposed in this thesis in order to have a fast heuristic.

2.6 Other Definitions of Critical Nodes

Many different definitions of "critical node" have been proposed in the literature due to the variety of viewpoints in defining network vulnerability. Various researchers defined critical nodes differently in their work, but they all share the meaning that those nodes are somehow rare and different than other nodes in the network, and they investigated the importance of these nodes in terms of disconnecting the networks. They used different objective functions such as the size of largest component or the average shortest path length to evaluate amount of disconnectivity.

Crucitti et al. [23] defined the critical nodes based on the degree centrality. To study the efficiency of scale-free complex networks, they attacked the benchmark networks (generated by the Barabasi-Albert [9] and Klemm-Eguiluz [52] models) based on the node degrees and then calculated the vulnerability of those networks to attacks. The results showed that scale-free networks are more resilient to random node removals than in random networks, but they are fragile to attacks on nodes with highest degrees. Albert et al. [2] studied the attack tolerance of communication networks such as the Internet, social networks, and cells based on removing nodes with higher degree centrality values. The results on real-world networks indicated higher tolerance against random node failures in scale-free networks than in exponential networks where most of the nodes have

approximately the same degree [2].

Later, Crucitti et al. [24] studied the vulnerability of complex networks based on three different definitions of criticality of the nodes: degree centrality, betweenness (load)[1] of the nodes, and recalculated betweenness (recalculated load)[2] of the nodes. These measures were tested on benchmark networks generated by Barabasi-Albert [9] and Erdös-Renyi [28] network models. The results of experiments showed that the networks are more vulnerable to attacks based on recalculated betweenness measure than other two measures.

Nardelli et al. [58] used the term “the most vital node" (MVN) instead of “critical node". They defined the most vital node as a node whose removal from the graph causes the largest increase of the distance between two specified nodes r and s. The study was to design a fast algorithm to determine the most vital node in a shortest path between two given nodes r and s [58]. They proposed an algorithm with time complexity $\mathcal{O}(|E|+|V|\log|V|)$, which is faster than the trivial solution of recalculating the shortest path between r and s after removal of each node from the shortest path that has complexity $\mathcal{O}(|V||E|+|V|^2\log|V|)$ [58].

Jorgic et al. [46] defined critical nodes for connectivity in Ad Hoc networks. A node in an Ad Hoc networks is called critical if its removal breaks the network into two or more disconnected components. The approach presented by Jorgic et al. [46] finds local critical nodes defined as: v is a local critical node if its removal disconnects the k-hop neighbours of v (the variable k is defined by user). The experiments on networks generated by a random graph unit model showed high performance based on comparisons between local and global critical nodes (global critical nodes are cut vertices mentioned in Section 2.4.1). As mentioned in [46], these k-hop (local) critical nodes are not always the global critical nodes.

Sheng et al. [69] partitioned the nodes of a graph into three categories: global critical nodes, local critical nodes, and ordinary nodes. Global critical nodes have the same definition as cut vertices. Local critical nodes are the nodes whose removal disconnects their k-hop neighbours, and ordinary nodes are those that are neither global critical nor local critical. Sheng et al. [69] presented a distributed algorithm to determine local and global critical nodes in Ad Hoc networks, however the details about evaluating the algorithm on benchmarks were not given.

Karygianni et al. [50] defined critical nodes of networks as follows: “A critical node is a node whose failure or malfunctioning disconnects or significantly reduces the performance of the network (i.e. introduces unacceptably long alternative paths)". A trigger mechanism was used to do some test traffic on the Ad Hoc network instances by send-

[1] Betweenness of node v is the number of shortest paths (over all pairs of nodes of the network) that pass through that node, evaluated before any removal is performed.

[2] Recalculated betweenness has the same quantity as the betweenness except that shortest paths are recalculated every time a node is removed.

ing packets from different places to each other. The gained information about incoming and outgoing packets determines which nodes may be critical for the network based on the incoming and outgoing packets from each node. There is no absolute definition of a “critical node", and they are referred to as “suspicious critical nodes".

Wehmuth and Ziviani [77] proposed an approach to calculate the criticality value of any node in a graph in terms of network connectivity. The proposed approach calculates the criticality of any node v based on a localized spectral analysis on its h-hop neighbours [77]. The performance of the proposed methodology was evaluated on benchmark networks, and a real-world network was used to evaluate the feasibility of the proposed approach on large networks. The benchmark networks were generated by Barabasi-Albert [9] and Erdös-Renyi [28] network models. The results of comparisons between their proposed approach and degree centrality on BA networks showed that both approaches perform similarly. In ER networks, the proposed methodology identified the correct critical nodes since in some cases some of the critical nodes were not the highest degree nodes. An internet router level network with 190,000 nodes was used as a real-world network to evaluate the feasibility of the proposed methodology to compute the critical nodes that have the most impact on the network connectivity. Wehmuth et al. [77] indicated that their approach is feasible to be used on the tested real-world network that has 190,914 nodes, however the runtime of the proposed algorithm was not reported.

Borgatti [17] defined the critical nodes as a set of nodes whose removal from the network results in maximum disconnectivity in the residual graph. As stated in [17], the previous centrality measures are not optimal for this problem and new heuristics need to be designed. The Borgatti's work was the start of the critical node detection problem (CNDP) studied in this thesis. Aruleslvan et al. [4] provided the formal definition of the CNDP and the proof of its NP-completeness.

2.7 Previous CNDP Work

After the CNDP [4] was formally defined in 2008, various approaches for this problem were proposed. They are highlighted here.

Arulselvan et al. [4] designed their heuristic for the CNDP based on maximum independent sets[3]. The proposed approach was tested on benchmark networks generated by a Barabasi graph generator and also a real terrorist network [4]. The size of the generated benchmark networks is in the range of 75 to 150 nodes, and the terrorist network contains 62 nodes. The runtime of the proposed heuristic is less than two seconds in all tested networks. The results of comparisons between the IP model and proposed ap-

[3] In a graph $G = (V, E)$, a maximum independent set (MIS) $M \subseteq V$ is a set of vertices that is not a subset of any other independent set: every edge of the graph G has at least one endpoint not in M, and every vertex not in M has at least one neighbour in M.

proach showed that the proposed approach was able to find the optimal solution and it found the solution much faster than the IP model [4]. Moreover, a local search was added to the methodology in order to improve the quality of the solution of the heuristic.

Recently, Arulselvan et al. [6] modified their heuristic for the Cardinality Constrained-CNP (CC-CNP). The CC-CNP has a limit on the maximum allowable connectivity index[4] for any node in a graph, the objective is to minimize the number of nodes to be deleted with considering the restriction on the connectivity index. Hence, the cardinality of each disconnected component in the induced subgraph graph $G(V \setminus L)$ must be less than the given connectivity index. They also proposed a GA solution for the CC-CNP [6]. The heuristic, the GA solution, and the IP model were compared on a terrorist network and some benchmark networks given at [4]. The terrorist network contains 62 nodes, and the size of the benchmarks is in the range of 20 to 150 nodes. The computational time of both proposed GA and heuristic was in seconds for all tested networks. The results on the terrorist network and benchmarks showed that the solutions of the heuristic and GA solution are close to the optimal solution and they are fairly comparable to each other.

Summa and Grosso [26] investigated the CNDP over trees in different situations of node weight w_i and edge cost c_{ij}, where w_i is the weight of removing a node i from network and c_{ij} is the cost of connection between i and j. Therefore, the objective of the CNDP given at Eq. (2.1) can be reformulated as:

$$\underset{L \subseteq V}{\operatorname{argmin}} \sum_{i,j \in (V \setminus L)} c_{ij}(G(V \setminus L)), \tag{2.24}$$

subject to $\sum_{i \in L} w_i \leq k$ where k is the given number of critical nodes to be deleted.

Arulselvan et al. [4] proved that in general networks ($c_{ij} = 1$ and $w_i = 1$ for any node) the complexity of the CNDP is NP-complete. Summa and Grosso [26] proved that the CNDP is still NP-complete over trees when the cost of connection between pairs of nodes is different ($c_{ij} \geq 0$). They also proved that the CNDP is solvable in polynomial time over trees when all connections between pairs of nodes have unit cost ($c_{ij} = 1$). Two dynamic programming approaches were proposed for the unit edge costs with unit node weights ($w_i = 1$) or arbitrary node weights ($w_i \geq 0$) [26]. Moreover, it was proved that the computational complexity of dynamic programming approaches for $c_{ij} = 1, w_i = 1$ and $c_{ij} = 1, w_i \geq 0$ situations are $\mathcal{O}(|V|^3 k^2)$ and $\mathcal{O}(|V|^7)$ [26], respectively.

Recently, Ventresca [72] proposed a simulated annealing (SA) strategy and a population based incremental learning (PBIL) approach for the CNDP. The heuristics were evaluated on benchmark networks generated by four different network models. The size of generated benchmarks is in the range of 250 to 5000 nodes. the average computational time of approaches vary in different benchmarks, and the runtime of approaches is in the range of 38 to 3515 seconds on all tested benchmarks. The SA has lower run-

[4] The connectivity index of a vertex is defined as the number of vertices reachable from that vertex.

time than PBIL in all benchmarks. The results of the comparisons between the SA and PBIL based on Cohen's d-statistics showed that PBIL outperforms SA in all benchmarks [72]. The results of heuristics were also compared to the best results of random sampling, which indicated that the mean results of the SA are less desirable than the best random sampling results in most of the cases while it was vice versa for the comparisons between the best random sampling and the mean results of the PBIL [72].

As discussed here, only a few solutions have so far been proposed for the CNDP. Furthermore, these methodologies have only been tested on small networks, with one previous work considering up to 5000 sized networks [72]. The aim of this thesis is to further contribute to the CNDP by proposing efficient heuristics feasible for larger complex networks ranging from thousands to millions of nodes.

2.8 Related Work to The CNDP

One of the problems similar to the CNDP is graph partitioning [51]. The problem is to partition the input graph G into r subgraphs of predefined size so that the number of edges that lie between them (the cut size) is minimal [32]. As stated in [32, 65], most variants of the graph partitioning problem are NP-hard, and therefore heuristics need to be developed to find good answers for these problems. The proposed partitioning problem is called the r-way partitioning of a graph [51]. The 2-way partitioning problem is called minimum bisection problem and it is also an NP-hard problem [32].

One of the earliest approaches proposed for the 2-way partitioning (minimum bisection) problem is the Kernighan-Lin algorithm [51]. The procedure optimizes a function Q that represents the difference between the number of edges that lie in the clusters and the number of edges between the clusters. This algorithm starts by making an initial partitioning of the graph into two clusters of predefined sizes, where the initial partitioning can be random or based on some information gained from the structure of the graph. Then, subsets of nodes that consist of an equal number of nodes are swapped between two clusters in order to maximize Q, and this procedure iterates until no more swapping can be done. The computational complexity of this algorithm is $\mathcal{O}(|V|^2 \log |V|)$. As mentioned in [32], the performance of the Kernighan-Lin algorithm depends on the selection of initial clusters, and the Kernighan-Lin algorithm is typically used to improve on the clusters found by other methods.

Another method for the minimum bisection problem is the spectral bisection method [10]. This method is based on using the eigenvalues of the Laplacian matrix[5]. The method calculates the eigenvalues of the Laplacian matrix and then selects the nodes whose corresponding value in the eigenvector of the second eigenvalue λ_2 has the same

[5]The Laplacian matrix Z is defined as $Z = D - A$, where A is the adjacency matrix of the graph and D is the degree matrix, which is a diagonal matrix and each diagonal entry of a row i is the degree of node i.

sign (negative or positive) and puts them in the same cluster [10]. The computational complexity of this algorithm is $\mathcal{O}(|V|^3)$, which is dominated by complexity of calculating all eigenvalues of the Laplacian matrix.

Hendrickson et al. [41] proposed a multilevel algorithm for the r-way partitioning problem. The size of the graph is reduced at each stage by removing vertices and edges from the graph and partitioning the remaining graph, then mapping back to the original graph. Hendrickson et al. [41] compared their approach to other partitioning methods on some graphs such as Hammend mesh, ocean mesh, etc. and concluded that their approach is as good as the best of other approaches and much faster than spectral partitioning approaches.

Pothen [65] presented a literature review of previous works on different classes of graph partitioning such as spectral partitioning, geometric partitioning, and multilevel algorithms. The graph partitioning problems with the objective of having r partitions of roughly the same size and minimizing the number of edges lying between clusters are similar to the objective of the CNDP, which is producing components that the variance between their cardinalities is minimized by removing k nodes from the graph. In the CNDP, the objective is to have components with least possible variance between their cardinalities and have as many components as possible [4], while the number of clusters in graph partitioning problems should be predefined [32].

Another interesting problem related to the CNDP is the maximum cut problem (MAX CUT). Karp [49] introduced the MAX CUT problem as follows: In a graph $G = (V, E)$ where each edge (i, j) has a non-negative weight $w_{ij} \geq 0$, the problem is to find a subset $S \subseteq V$ of nodes such that the summation of weights of edges lie between S and $V - S$ is maximized. It was proved that the decision version of this problem is NP-complete [49]. The decision version of the maximum cut problem is to determine the eligibility of a solution S where the summation of weights of edges that lie between S and $V - S$ is greater than or equal to a given value W. The set of edges that lie between S and $V - S$ is called “the cut" in the literature. Garey et al. [37] proved that the MAX CUT problem is also NP-complete in unweighted graphs ($w_{ij} = 1$ for any edge $(i, j) \in E$), where the summation of weights in the cut is equal to the number of edges that lie between S and $V - S$.

Since it was proved that the maximum cut problem is NP-complete, α-approximation algorithms need to be developed, which means that the quality of solution of the algorithm is at least α times lower than the optimal value. Sahni et al. [66] proposed a 0.5-approximation algorithm for the maximum cut problem on unweighted graphs, which can also be used for weighted graphs. Their algorithm iterates on all nodes and checks whether moving a node u from its group to another group maximizes the weight of cut. The computational complexity of the proposed approximation algorithm is $\mathcal{O}(|V| + |E| + k)$ [66], where k is the number of groups the nodes are to be partitioned, which is $k = 2$ for the MAX CUT problem. Crescenzi et al. [22] proved that the MAX CUT problem

on unweighted graphs is as hard to approximate as in weighted graphs. Therefore, the found approximation upper-bounds for a version of the problem can also be considered for another version.

Goemans and Williamson [38] proposed a randomised approximation algorithm for the MAX CUT problem with approximation ratio $\alpha \approx 0.878$, which is the best known approximation and if the unique games conjecture is true, it would be the best possible approximation. Papadimitriou and Yannakakis [64] showed that there exists a constant $c > 0$ such that the polynomial time c-approximation for the MAX CUT problem is NP-hard. Bellare et al. [12] proved that finding an approximation for the MAX CUT problem with ratio better than $\frac{71}{72} \approx 0.986$ is NP-hard. Later, Trevisan et al. [71] improved the upper-bound of the ratio of approximation and proved that no polynomial time approximation for the MAX CUT problem can be found with ratio better than $\frac{16}{17} \approx 0.941$ unless $P = NP$.

A version of the MAX CUT problem is called MAX k-CUT, which is to have a partition S with k groups, where the summation of the weights of edges that lie between the groups is maximized. The approximation algorithm proposed by Sahni et al. [66] can be used for the MAX k-CUT problem as well, which has $\left(1 - \frac{1}{k}\right)$ approximation ratio. Papadimitriou and Yannakakis [64] proved that it is NP hard to c-approximate the MAX k-CUT problem on unweighted graphs for a constant $c > 0$ in any $k \geq 2$. Frieze and Jerrum [36] stated in their paper: "there can be no polynomial time approximation scheme for MAX k-CUT, for any $k \geq 2$, unless $P = NP$", which does not mean that the MAX k-CUT problem in weighted networks is NP-complete. An approximation algorithm for the MAX k-CUT was proposed in [36], which is an extension from the work of Goemans and Williamson [38]. The ratio of this approximation algorithm is $\left(1 - \frac{1}{k} + 2k^{-2}\ln k\right)$ [36]. Kann et al. [47] proved that no polynomial time approximation algorithm for the MAX k-CUT problem can be found with approximation ratio better than $\left(1 - \frac{1}{34k}\right)$ unless $P = NP$.

Arulselvan et al. [4] mentioned in their paper that an approximation of the MAX k-CUT problem can solve the problem of only maximising the number of components in the residual graph after deleting k critical nodes. However, this is not enough for the objective of the CNDP, which is to maximize the number of components and also minimize the variance of the size of the components in the residual graph [4].

Chapter 3

Methodology

This chapter describes the proposed algorithm for the CNDP. The algorithm ranks the nodes based on the local information of nodes such as the degree or vertex similarity value in order to find the most critical nodes to be removed from the graph. A key challenge and main contribution is to design an efficient ranking function in order to determine a suitable set of the k highest ranked nodes to remove from graph. In addition, a post-processing procedure to boost the quality of solution of the heuristic after the nodes are ranked by a ranking function is presented.

3.1 Depth-First Search Based Methodology

The depth-first search based heuristic (*DFSH*) that collects necessary node attributes during a DFS pass over all nodes of the input graph G and assigns a rank to each node is proposed. The DFS helps to gain important information about the nodes of graph such as indicating cut vertices and bridges, which is the reason of using it instead of other search procedures like breadth-first search (BFS). Algorithm 1 depicts the main steps of the proposed methodology. The DFS criterion for exploring all the nodes in a graph and selecting the k highest ranked nodes (line 15) is trivial, while ranking each node during the DFS search (line 7) is the key challenge since the quality of solution of the heuristic is dependant on the nodes selected by the ranking function. Because the CNDP is an NP-complete problem, designing a ranking function for an optimal solution is computationally intractable. Since one main goal is to develop fast heuristics, restrictions on the computational complexity limit the use of calculating time-consuming node attributes (e.g., betweenness and closeness) in calculating the ranks of nodes. Thus, node attributes that need at least $\mathcal{O}(|V|^2)$ run time for calculation are not employed. In addition, finding suitable local vertex properties to extract from the network is also a major concern and a challenge, because the selected vertex properties must be helpful to find the most critical nodes regarding the objective of the CNDP. The details of the search strategy, ranking function, and selection mechanism are described.

Algorithm 1 DFS based algorithm for the CNDP

Require: $G = (V, E)$ and starting node v

```
1:  Stack S := ∅
2:  push(S, v)
3:  while S is not empty do
4:      u := pop(S)
5:      if u is not explored then
6:          label node u as explored
7:          update the ranking value of u
8:          for w ∈ Γ(u) do
9:              if w is not explored then
10:                 push(S, w)
11:             end if
12:         end for
13:     end if
14: end while
15: select k nodes of the highest ranks
```

3.1.1 Searching The Network

An efficient searching technique was needed to visit all nodes of the network and gather information about the network and each explored node. One of the basic and fast algorithms to explore a graph is the depth first search, which has a run time $\mathcal{O}(|V| + |E|)$ and space complexity $\mathcal{O}(|E|)$. The DFS algorithm can be used to determine all cut vertices and bridges based on the information gained about the DFS tree G_T of a graph G [21]. Removing cut vertices or bridges from a given network results in immediate disconnection in the network and consequently a decrease in the objective value of the network. So these vertices and edges are considered when assigning a vertex its rank.

3.1.2 Ranking The Nodes

All of the nodes in a graph G need to be ranked, where the rank of each node represents its importance of being removed from graph. The idea is to assign scores to each node during the DFS search based on the obtained local information from the nodes (e.g., cut vertices, vertex similarity values, and node degrees). A ranking function assigns higher ranks to nodes that are considered more important according to the objectives of the CNDP by considering a weighted combination of the local information gained about each node.

3.1.3 Selection Mechanism

All the nodes of a given input graph G are scored based on local information obtained during the DFS search. The next step is to select k nodes of the highest ranks. A possible

way to accomplish this is to add each node to a priority queue and then extract the k nodes from that queue. The computational complexity of adding $|V|$ nodes to a binary heap is $\mathcal{O}(|V|\log|V|)$, and the extraction of k nodes is $\mathcal{O}(k\log|V|)$, since the complexity of deleting the maximum member of heap is $\mathcal{O}(\log|V|)$. Another way to extract k nodes of the highest ranks is to put all nodes in an array and sort them, which has $\mathcal{O}(|V|\log|V|)$ complexity [21], and then extracting k nodes of the highest ranks is $\mathcal{O}(k)$. The second approach is used here.

3.2 Ranking Function

A ranking function assigns ranks to each node in the graph. The ranking function should be designed based on the objective of the CNDP. The ranking function (*RANKH*) for a node $i \in V$ is formulated as follows:

$$RANKH(i) = \sum_{j\in\Gamma(i)} \Big(\big(1-\beta(i,j)\big)\Big(w_1\big(1-V_S(i,j)\big)\Big) + \beta(i,j)\Big(\tau(i,j)\big(w_2\frac{deg(i)}{\Delta(G)+1}\big) + \big(1-\tau(i,j)\big)\big(w_3\frac{deg(i)}{\Delta(G)+1}\big)\Big)\Big) + w_4(1-\lambda_{i,G}), \tag{3.1}$$

where

$$\beta(i,j) = \begin{cases} 1 & \text{if } V_S(i,j) \text{ is zero (i.e., edge } (i,j) \text{ is either a bridge or a local bridge,)} \\ 0 & \text{otherwise,} \end{cases} \tag{3.2}$$

$$\tau(i,j) = \begin{cases} 1 & \text{if edge } (i,j) \text{ is a bridge,} \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

$$\lambda_{i,G} = \begin{cases} \dfrac{f(G\setminus\{i\})}{f(G)} & \text{if node } i \text{ is a cut vertex,} \\ 1 & \text{otherwise,} \end{cases} \tag{3.4}$$

$\lambda_{i,G}$ calculates the impact of removing i from G and $0 \le \lambda_{i,G} \le 1$, where function $f(G)$ is the objective value of graph G given in Eq. (2.3). Consequently, $(1-\lambda_{i,G})$ is higher for a cut vertex i whose removal results lower objective value of the induced subgraph $G(V\setminus\{i\})$. $\Delta(G)$ is the maximum degree of the nodes of G, and $V_S(i,j)$ is the Sorensen-Dice similarity value of node i and its neighbour j introduced in Sub-section 2.4.2. Preliminary experiments showed that lower vertex similarity values are better in terms of the objective of the CNDP (the two nodes have less vertices in common), and therefore the $(1-V_S(i,j))$ is used to give higher scores to nodes having lower vertex similarity. The weights w_1, w_2, w_3, and $w_4 \ge 0$ will be empirically determined based on statistical experiments on dif-

ferent combinations of these weights to find the most suitable set of weights (detailed information is given in Section 3.2.1).

Thus, the ranking function given above is a linear combination of local properties calculated for each node. A score is calculated and added to a node for each of its incident edges, while the cut vertices receive an extra score ($w_4(1-\lambda_{i,G})$). The scores calculated for bridges, local bridges, and regular edges (edges with vertex similarity greater than zero) may be different from each other, the difference depends on weights w_1, w_2, and w_3, respectively. Therefore, the more incident edges to a node yields a higher score for that node. Some researchers examined the impact of deleting high degree nodes on different benchmark networks [23, 24] and stated that scale-free networks are vulnerable to the deletion of high degree nodes. This thesis differentiated between different kinds of edges (bridges, local bridges, and regular edges) and assigned different scores to each. The endpoints of a bridge or a local bridge receive a score based on their degree multiplied by different weights (w_2 for local bridges and w_3 for bridges). The endpoints of a regular edge receive a score based on their calculated vertex similarity value, which is multiplied by weight w_1. The purpose of designing the *RANKH* function with undetermined weights w_1 through w_4 was to give flexibility to the ranking function in order to be adopted to different network models with various topologies.

Since the bridges and cut vertices of G are determined during the DFS search, the computational complexity of determining all cut vertices and bridges is $\mathcal{O}(|V|+|E|)$. Therefore, the worst-case complexity of *RANKH* is dominated by the complexity of calculating vertex similarity values. In the following theorem the complexity of *RANKH* for calculating the rank of a node in the input graph is given based on the maximum degree of any node in G.

Theorem 3.2.1. *RANKH has worst-case complexity $\mathcal{O}(\Delta(G)^2)$ for calculating the rank of any node v in graph G.*

Proof. The complexity of calculating the common neighbours between any node v and one of its neighbours is $\mathcal{O}(deg(v))$ and we have

$$deg(v) \leq \Delta(G), \tag{3.5}$$

where $\Delta(G)$ is the maximum degree of any node in G. Therefore, the complexity of calculating vertex similarity between any node and one of its neighbours in the graph is $\mathcal{O}(\Delta(G))$, and consequently the complexity of calculating vertex similarity for all adjacent neighbours of a node is $\mathcal{O}(\Delta(G)^2)$. □

In the worst case, graph G is fully connected and therefore the degree of each node is equal to $|V|-1$, and also $\Delta(G)=|V|-1$, which means that the complexity for calculating all neighbours of any node is $\mathcal{O}(|V|^2)$. However, most real-world networks are sparse

graphs ($|E| \ll |V|^2$). The average degree of nodes in real-world networks may help to highlight the difference of using the *RANKH* in real-world networks than in complete graphs. Reka et al. [1] tested many real-world networks, and their results showed that the average degree of nodes in real networks is a very small number (less than a hundred) compared to the network size (ranging from thousands to millions of nodes). For example, the average degree of the nodes in a phone call network with 53 million nodes was 3.16 [1].

The pseudo-code of an iterative DFS and *RANKH* that calculates the vertex similarity value between endpoints of any edge is shown in Algorithm 2.

Algorithm 2 Iterative DFS including calculation of vertex similarity

Require: Graph $G = (V, E)$ and starting node v
1: Stack $S := \emptyset$
2: push(S, v)
3: **while** S is not empty **do**
4: $u :=$ pop(S)
5: **if** u is not explored **then**
6: label u as explored
7: **for** $w \in \Gamma(u)$ **do**
8: calculate $|\Gamma(u) \cap \Gamma(w)|$
9: update ranks of u and w
10: **if** w is not explored **then**
11: push(S, w)
12: **end if**
13: **end for**
14: **end if**
15: **end while**
16: select k nodes of the highest ranks

As defined in [21], the cut vertices and bridges are calculated by constant-time operations during the DFS search. The complexity of calculating the ranks of all nodes in graph G is given in the following theorem. It should be noted that the input graph is assumed to be connected and therefore $|E| \geq |V| - 1$.

Theorem 3.2.2. *Algorithm 2 has complexity* $\mathcal{O}(|E|\Delta(G) + |V|\log|V|)$.

Proof. Since the while loop on lines 3-15 executes for each node of the graph G only once, it requires $\mathcal{O}(|V|)$. For any node $v \in V$, the loop on lines 7-13 takes time $\mathcal{O}(|\Gamma(v)|\Delta(G))$, since line 8 is an $\mathcal{O}(\Delta(G))$ operation. The loop on lines 7-13 executes for each node in the graph. Since

$$\sum_{v \in V} |\Gamma(v)|\Delta(G) \leq \Delta(G) \sum_{v \in V} |\Gamma(v)| = \mathcal{O}(|E|\Delta(G)),$$

the cost of executing lines 3-15 takes $\mathcal{O}(|V| + |E|\Delta(G))$. Based on the assumption that the graph is connected and therefore $|E| \geq |V| - 1$, the complexity of lines 3-15 is $\mathcal{O}(|E|\Delta(G))$.

As stated in Section 3.1.3, the complexity of selecting k nodes of the highest ranks is $\mathcal{O}(|V|\log|V|)$. Therefore, the overall complexity of algorithm 2 is $\mathcal{O}(|E|\Delta(G)+|V|\log|V|)$. □

In the worst case, the graph is fully connected and the degree of each node is equal to $|V|-1$, and also $\Delta(G)=|V|-1$. Hence, the complexity of Algorithm 2 (*DFSH*) would be $\mathcal{O}(|E||V|+|V|\log|V|)$. As mentioned earlier in this section, many sparse real-world networks were observed that had very low average degree of nodes [1], and therefore the complexity of *DFSH* in real networks is expected to be far from the worst case.

3.2.1 Weight Tuning Procedure

As discussed in Section 3.2, the weights $w_1,\ldots,w_4$ in *RANKH* need to be established. Based on the fact that the used node characteristics may have different influence on the selection of nodes for various network structures, the goal was to calculate the weights of *RANKH* for different network structures and k-values. As an example, when only 5% of nodes of a network are cut vertices, even if all cut vertices are selected, other node characteristics must be used and may be even play a more important role than cut vertices when $k>5\%$. Therefore, the experiments on weights were designed to calculate the best combination of weights for k-values in the range of 1% to 50% with a 10% step for a network due to the fact that the role of node characteristics may change after changing the k-value.

In order to determine the best combination of weights for different network structures and k-values, a couple of network models were used to generate networks of different sizes. where each twenty network instances have the same size. The procedure of generating proper network instances from a network model and calculating the best set of weights for different k-values are described in this section. As discussed in Chapter 2, different networks have varying topologies and characteristics, and it is not practical to use one set of weights for all networks. For example, the scale-free networks have many nodes of low degrees and the likelihood of having cut vertices in these networks is higher than in small world networks. However, the nodes in small world networks are more connected to each other and especially to their closer neighbours (the clustering coefficient is higher [75]). Hence, it is unlikely to have cut vertices or bridges in small world networks. In order to determine the best sets of weights for different network topologies, four different network models (introduced in Section 4.1) were used to generate benchmark data, and then the effect on the objective value after removing nodes selected by *DFSH* for all tested combinations of weights are calculated and for each k-value the set of weights that results in the lowest objective value is reported.

The range of numbers assigned to each of the weights w_1 to w_4 was $[0,1]$ with a 0.15 step. An analysis on the sensitivity of calculated weights were done which indicated that

not much improvement can be observed by making smaller changes to selected weights (the results are shown in Section 4.2). The lowest value 0 for a weight indicates that its corresponding node attribute is neutral in assigning ranks to nodes. When the highest value for a weight resulted in better objective values than lower values, the range of weights expanded until no further statistical improvements could be observed on the objective values of the benchmarks. The procedure for generating benchmarks by a network model and determining the best set of weights for each k-value based on statistical comparisons are given below.

1. Generate 100 different network sizes ranging from 100 to 25,000.
2. For each network size, generate 20 network instances of that size.
3. Calculate the objective values of all network instances after deleting k selected nodes based on different combinations of weights for *RANKH*. The k-value is in the range of 1% to 50% of network size with 10% step.
4. For each network size do:
   (a) Find the set of weights such that its average objective value for a k-value on twenty network samples is the least among other sets of weights, and give one score to that set of weights.
   (b) Do a t-test between the best set of weights and other sets, and give one score to each set where $p\text{-}value > 0.05$ (i.e., the best set of weights is not significantly better than the compared set of weights).
5. Report the set of weights with the highest score for each k-value. Since there are 100 network sizes in a benchmark suite, the highest possible score for a set of weights is 100.

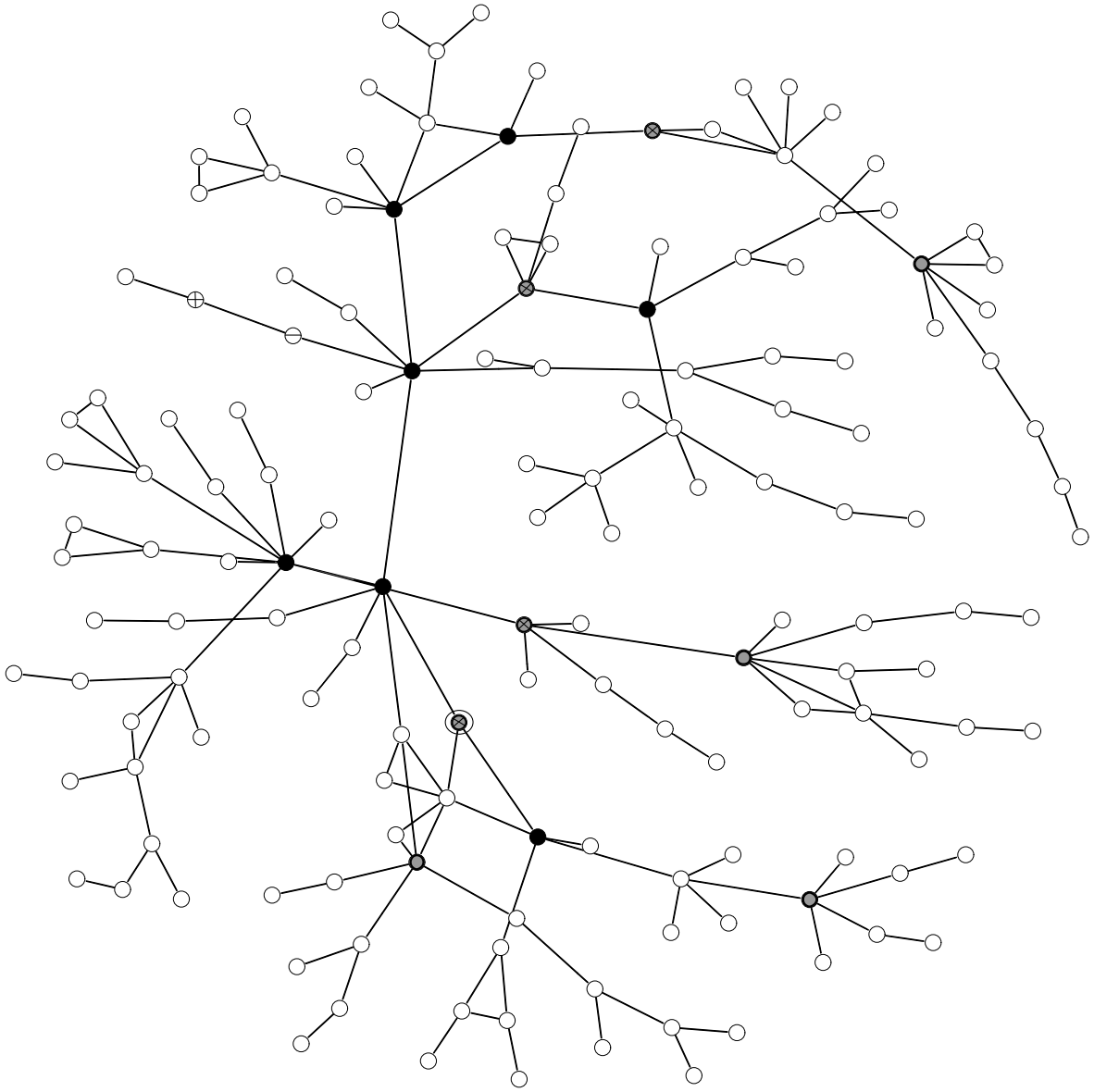


Figure 3.1: A Forest Fire network sample with 150 nodes, where the shaded nodes represent the solutions from the optimal answer (calculated by the IP formulation) and *DFSH* when $k = 11$. Black nodes are in both solutions, while the grey nodes are selected in the optimal solution and grey nodes with multiplication notation (×) are selected in *DFSH*.

3.3 Post-processing Procedure

In order to improve the quality of the solution of the DFS-based heuristic, a post-processing procedure is proposed. An example of an optimal solution obtained from the IP formulation (Section 2.2.1) and *DFSH* on a Forest Fire network instance with 150 nodes and $k = 11$ is shown in Figure 3.1. The black nodes are those selected by both approaches. The grey nodes and grey nodes with multiplication notation ($\times$) represent nodes selected by only the optimal solution and *DFSH*, respectively. The grey nodes that are selected by the optimal solution are not connected to any of the black nodes, while all the grey nodes with multiplication notation are connected to black nodes in this case. Consequently, a post-processing procedure was proposed to deselect any node $v \in L$ such that the ratio of its neighbours in set L to its degree is greater than a given threshold. The idea behind this post-processing procedure is that some nodes selected by *DFSH* may have many selected neighbours and after removing their neighbours their removal may not be necessary any more. An example of the case where deselecting nodes definitely improves the objective value of the graph is when all neighbours of node $v \in L$ are also in set L. Node v will be a node of degree 0 after all of its neighbours are removed from the graph. Therefore, removing node v is not a good idea any more since the objective value of the graph does not decrease after deletion of an isolated node. In order to determine which nodes in set L should be deselected, a threshold θ is used so that the nodes meeting the threshold during the post-processing procedure will be deselected. Therefore, the value of θ needs to be determined for different network topologies and k-values based on experimental results. An example is shown in Figure 3.1, where the double circled node in grey color with multiplication notation has two selected black color neighbours and its degree is 3. The ratio value for this node is $\frac{2}{3} = 0.66$, and it will be deselected when the threshold θ is less than 0.66. In many cases of the *DFSH*, the high ranked nodes had neighbours of high ranks, and since removing the neighbours of a high ranked node has influence on its rank, it is important to consider the ratio of selected neighbours for each selected node in set L.

After the ranks of all nodes were calculated by *DFSH*, the post-processing procedure was performed on the k selected nodes, the pseudo-code of the post-processing steps is given in Algorithm 3.

For each node $v \in V$ the number of its neighbours that are in set L need to be determined. This is done in lines 1-5 of Algorithm 3 by increasing the number $\Phi(w)$ for all neighbours of the nodes in set L, where $\Phi(w)$ is the number of neighbours of a node w that are in set L. At the beginning of the procedure $\Phi(w) = 0$, $\forall w \in V$. Then, by starting from the lowest rank node in L (line 6), all the selected nodes in set L are checked to determine whether they should be deselected. Each node where the ratio of its neighbours in set L to its degree is greater than a given threshold (named θ in line 7) will be removed from set L. The value $\Phi(w)$ for all neighbours $w \in \Gamma(u)$ of a node $u \in L$ are decreased after

Algorithm 3 Post-processing procedure

Require: Graph $G = (V, E)$ and set L of k selected nodes

for $u \in L$ **do**
  for $w \in \Gamma(u)$ **do**
    $\Phi(w)$ ++
  end for
end for
for $u \in L$ (start from the lowest rank to the highest rank) **do**
  if $\frac{\Phi(u)}{deg(u)} \geq \theta$ **then**
    remove node u from set L
    for $w \in \Gamma(u)$ **do**
      $\Phi(w) - -$
    end for
  end if
end for
while $|L| < k$ **do**
  select next node $u \in V - L$ {start from the unselected node with the highest rank to the lowest rank}
  if $\frac{\Phi(u)}{deg(u)} < \theta$ **then**
    add node u to set L
    for $w \in \Gamma(u)$ **do**
      $\Phi(w)$ ++
    end for
  end if
end while

that node is removed. The nodes with the lowest ranks are checked to be deselected first since a high rank node may not be eligible to be removed from L any more after a few of its low rank neighbours in L are deselected. After the deselection procedure (lines 6-13) is complete, new nodes must be added to set L in order to maintain k nodes in set L. The selection procedure (lines 14-22) starts from the next highest ranked node u in $V-L$ and adds it to L if the ratio $\frac{\Phi(u)}{deg(u)}$ is less than threshold θ, since the aim is to have a set L that the ratio $\frac{\Phi(u)}{deg(u)}$ for any $u \in L$ is not greater than the threshold θ. The value $\Phi()$ for all neighbours of a newly selected node is increased in order to consider it for further selections. The complexity of Algorithm 3 is given in the following theorem.

Theorem 3.3.1. *Algorithm 3 has complexity* $\mathcal{O}(k\Delta(G)+|V|)$

Proof. The complexity of nested loops in lines 1-5 is equal to the summation of degrees of nodes in set L of k nodes. Since

$$\sum_{u\in L}|\Gamma(u)| \leq \sum_{u\in L}\Delta(G) \leq \Delta(G)\sum_{u\in L}1 = \mathcal{O}(k\Delta(G)),$$

the cost for executing lines 1-5 is $\mathcal{O}(k\Delta(G))$. The complexity of the loop in lines 6-13 is $\mathcal{O}(k)$ since it checks all nodes of L and $|L| = k$. The worst case for lines 14-22 is to check all nodes in $V-L$ (i.e., it does not terminate until the end of checking all unselected nodes), so the complexity of this part is $\mathcal{O}(|V|)$. The complexity of all parts is $\mathcal{O}(k\Delta(G)+k+|V|)$, and since $k \leq |V|$ the overall complexity is $\mathcal{O}(k\Delta(G)+|V|)$. □

In the worst case, the input graph is fully connected ($\Delta(G) = |V|$), and the k-value is equal to the size of the network ($k = |V|$). Hence, the complexity of Algorithm 3 is $\mathcal{O}(|V|^2)$.

As stated in Theorem 3.2.2, the complexity of *DFSH* is $\mathcal{O}(|E|\Delta(G)+|V|\log|V|)$. Since the complexity of the post-processing procedure proposed in Algorithm 3 is $\mathcal{O}(k\Delta(G)+|V|)$, using the post-processing procedure after ranking the nodes does not affect the asymptotic complexity of *DFSH* given in Algorithm 2. The post-processing procedure given in Algorithm 3 added to *DFSH* is called *DFSH-post*. The threshold θ needs to be determined for *DFSH-post* based on tuning experiments in the same manner as the procedure stated in Section 3.2.1. The tested threshold θ values are in the range of 0.1 to 1.0 by a 0.05 step. An analysis was done on the sensitivity of calculated thresholds which indicated that smaller steps does not affect the quality of solution of *DFSH-post*, the results are shown in Section 4.2.

3.4 Earlier Designed Approaches

Before designing the *RANKH* and post-processing procedure defined in Chapter 3, simpler ranking functions with less number of weights were developed and investigated. Further investigations on experimental comparisons indicated that more weights can help the ranking function to get better results in various network topologies, and consequently the *RANKH* function has more weights and is more complex than previous approaches. Moreover, two post-processing procedures were also developed and are discussed in Appendix A. Furthermore, the detailed experimental comparisons between earlier designed approaches and those proposed in sections 3.2 and 3.3 are given in Appendix A.

Chapter 4

Benchmarking

As mentioned in Section 3.2, the weights of the ranking function need to be established. This chapter discusses the models that were used to generate various benchmark problem instances. The aim of benchmarking is to determine the weights of the ranking function for a set of networks with the same topological properties, and then use that set of weights for unseen networks with topologies similar to the tested networks. *DFSH-post* is then compared to the centrality-based approaches based on calculated weights and θ value. Moreover, *DFSH-post* is compared to the results of the population based approaches, presented in [72], on small sized networks. The centrality-based approaches [23, 24, 35, 57] are compared to *DFSH-post*. An overview of the main steps of methodology is given in Figure 4.1.

4.1 Benchmark Network Models

4.1.1 Erdös-Renyi Model

The Erdös-Renyi (ER) model is a random graph model that was introduced in 1959 [28]. The ER model generates a graph $G_{n,p}$ by starting from a graph with n isolated nodes and adding an edge between each pair of nodes with probability p. The probability of a node in $G_{n,p}$ with degree k is

$$p_k = \binom{n-1}{k} p^k (1-p)^{n-1-k} \approx \frac{\langle k \rangle^k e^{-\langle k \rangle}}{k!}, \tag{4.1}$$

where $\langle k \rangle = p(n-1)$ is the average degree of $G_{n,p}$. The approximation of distribution is Poisson and becomes exact as $n \to \infty$ and $\langle k \rangle$ is a constant [59]. Fortunato and Castellano [33] stated that the ER graph model has no community structure since the probability of existence of any edge in a random graph is equal to other edges, so there is no preferential attachment between different groups of vertices in the graph.

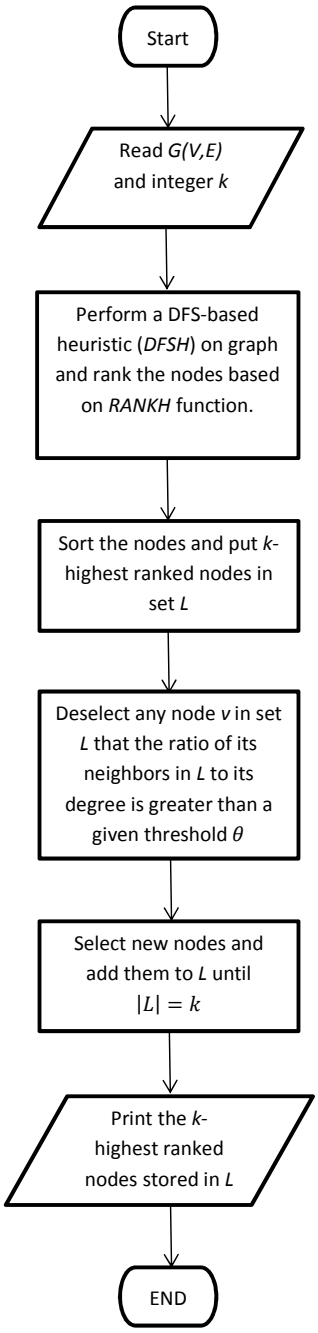


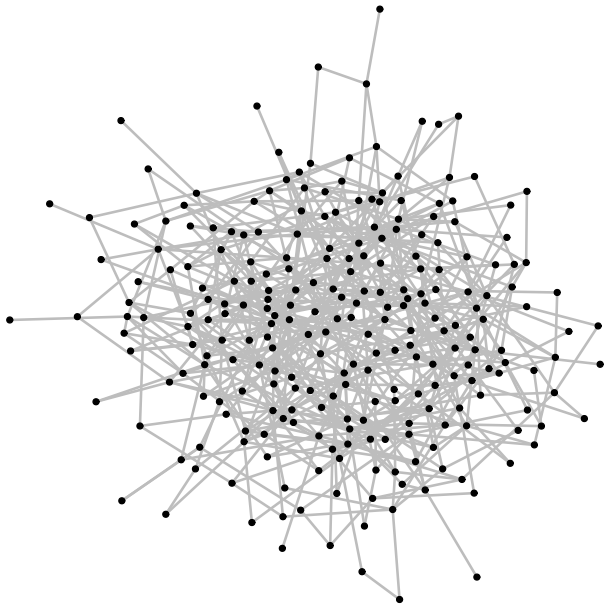
Figure 4.1: The main steps of the methodology.

Although the ER model produces graphs that do not have any of the common properties observed in complex networks (Section 2.3), it was used to produce benchmarks and compare the results of *DFSH* and the centrality measures in order to assess the quality of approaches on networks where no particular complex network attribute exists. An example ER network with 250 nodes and its log-log degree distribution are shown in Figure 4.2.

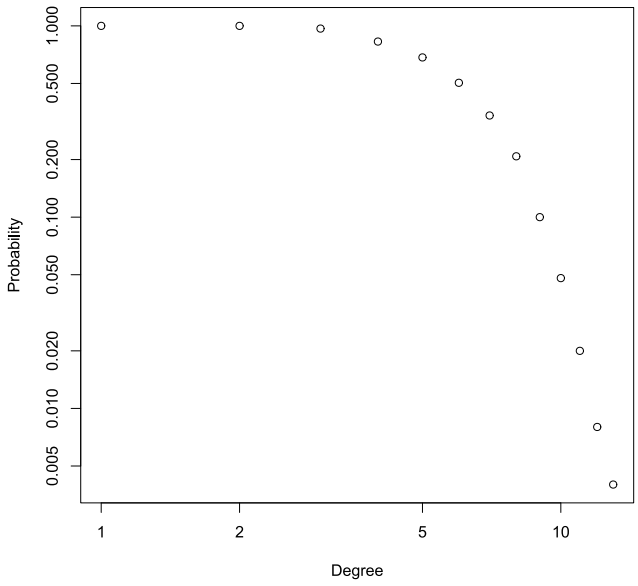
4.1.2 Watts-Strogatz Model

One of the common properties of complex networks is the small-world phenomenon, which was first pointed out by Milgram in 1967 and known as six degrees of separation [56]. The two aspects of small world networks are the low diameter of the network compared to its size and its high clustering coefficient, which is observed in many real-world networks (examples shown in [74] for social networks, metabolic systems, the Internet, etc.). The Watts-Strogatz (WS) model [75] aims to generate networks that have the two characteristics observed in real small-world networks. This model starts with a ring of n vertices and connects each vertex in the ring to all of its k nearest neighbours. Then, each edge is considered with probability p to rewire one of its endpoints to a randomly chosen node. Watts and Strogatz studied the effect of choosing p on the clustering coefficient and diameter of the network and stated that the best value of p is 0.05, which causes both high clustering coefficient and low diameter. An example WS network with 250 nodes, $k = 4$, and rewiring probability $p = 0.05$, along with its log-log degree distribution are shown in Figure 4.3(a).

As stated in [48], graphs generated by the Watts-Strogatz model have no community structure based on the definition of community structure from a link topology point of view, which is to have more intra-community links than inter-community links [32]. The WS networks should be considered hard to solve for the CNDP due to the density of links and the small-world property, i.e., a considerable number of nodes are required to be removed in order to increase the number of components in the induced subgraph $G(V \setminus L)$.

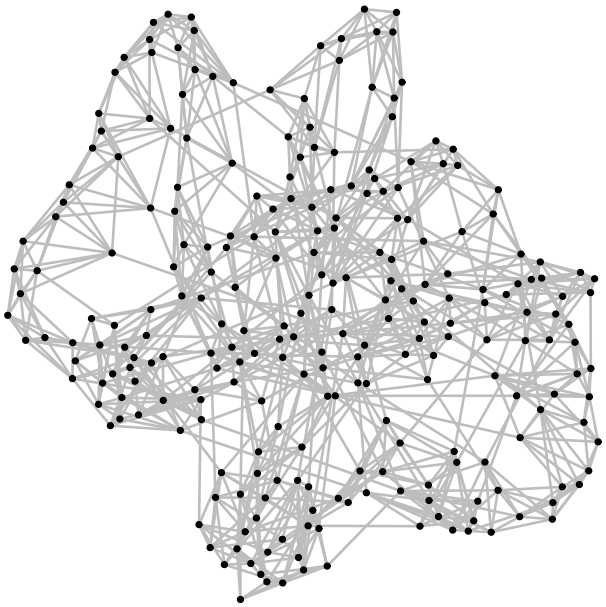


(a) An example ER graph $G_{n,p}$ with $n = 250$ and $p = 0.01$.

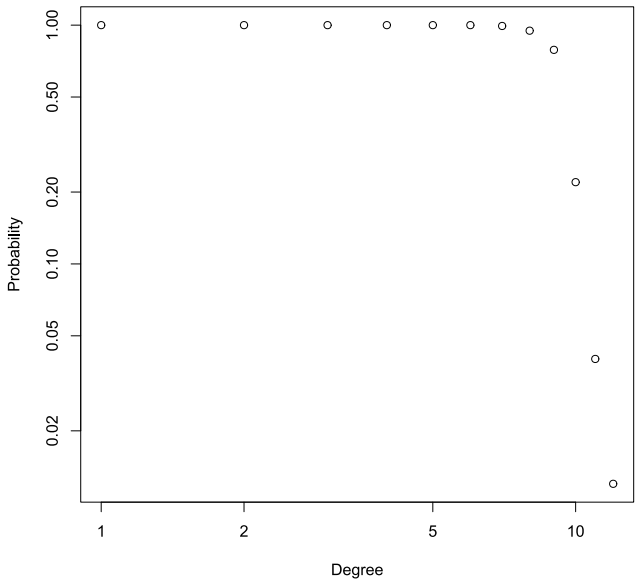


(b) The log-log degree distribution of an Erdös-Renyi network with 250 nodes.

Figure 4.2: An example Erdös-Renyi network and its log-log degree distribution.



(a) An example WS graph with $n = 250$, $k = 4$, and $p = 0.05$.



(b) The log-log degree distribution of a Watts-Strogtaz network with 250 nodes.

Figure 4.3: An example Watts-Strogatz network and its log-log degree distribution.

4.1.3 Barabasi-Albert Model

The scale-free property is one of the common properties of real-world complex networks, which can be observed in the Internet, World Wide Web, social networks, and airline networks [74]. The Barabasi-Albert (BA) model [9] produces networks with power-law degree distribution. Two mechanisms are used in the BA model in order to produce networks. The first mechanism is network growth, where the algorithm starts with m_0 nodes, and at each time step a node is added and connected to m existing nodes in the graph. The second mechanism is called preferential attachment, where the probability of connecting the new node u to any node v is related to the degree of v, which can be formulated as:

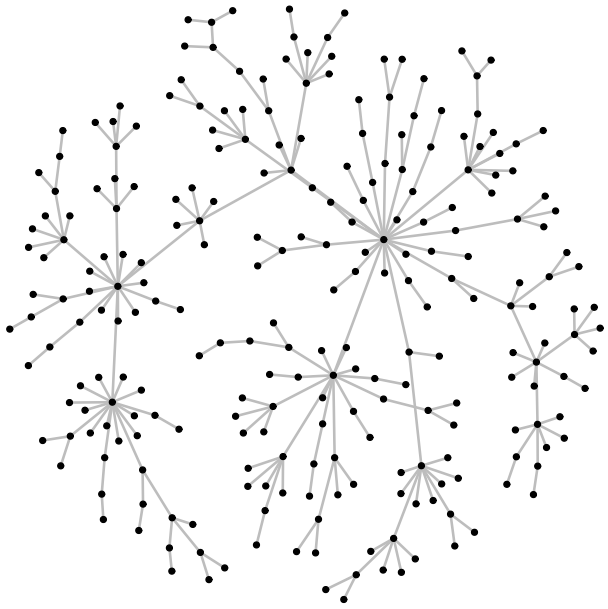
$$p((u,v) \in E) = \frac{deg(v)}{\sum_{w \in V} deg(w)}, \tag{4.2}$$

where $v, w \in V$ are pre-existing nodes in the graph. In other words, the new nodes are more probable to be linked to the existing nodes of higher degrees. The degree distribution of BA model [27] has been shown to have a power-law shaped distribution with exponent 3:

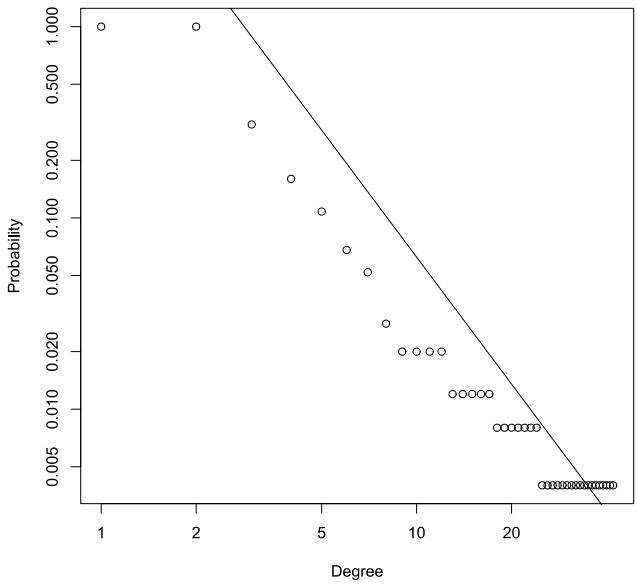
$$p_k = k^{-3}. \tag{4.3}$$

However, modifications on the model can lead to different power-law exponents [9]. Bollobas et al. [15] gave theoretical proofs about the sensitivity of defining the initial graph m_0 and its effect on the BA network properties such as clustering or average node degree, which was not mentioned in [9].

Since the BA model generates trees when $m = 1$, it is considered an easier problem for the CNDP as Summa et al. [26] proved that there is a polynomial time algorithm that yields an optimal solution for the CNDP in this case. The BA model with $m = 2$ was used to generate another benchmark suite, where no bridge exists due to the fact that each new node at a time step is added to two existing nodes creating a cycle in the graph. The benchmark suite generated by setting $m = 1$ is called *BA-m1*, and the one that contains networks generated by $m = 2$ is called *BA-m2* in this thesis. Figures 4.4 and 4.5 provide examples of BA networks when $m = 1$ and $m = 2$, respectively. Liu et al. [55] studied the community structure in BA networks. They calculated the Q value of the modularity algorithm proposed by Newman and Girvan [60] in order to measure the community quality of BA networks and stated that a value Q greater than 0.3 indicates significant community structure in a network. The results showed that the Q value in BA networks was always about 0.27, and therefore very weak community structure emerges. It is important to note that the preferential attachment is not the only way of generating scale-free networks, and other network models such as the copying model can do it [15].

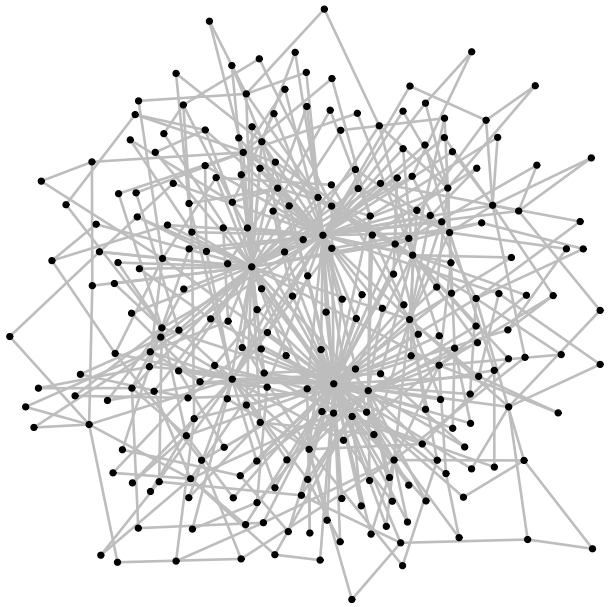


(a) An example BA network with 250 nodes and $m = 1$.

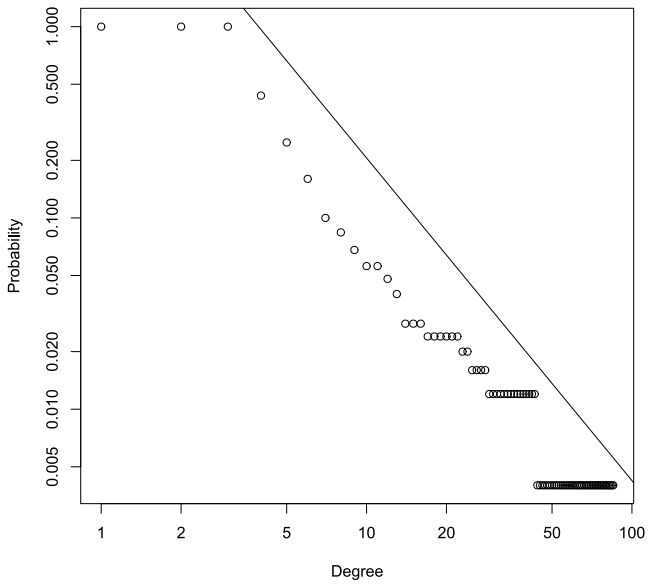


(b) The log-log degree distribution of a BA-m1 network with 250 nodes.

Figure 4.4: An example BA-m1 network and its log-log degree distribution.



(a) An example BA network with 250 nodes and $m = 2$.



(b) The log-log degree distribution of a BA-m2 network with 250 nodes.

Figure 4.5: An example BA-m2 network and its log-log degree distribution.

4.1.4 Forest Fire Model

The Forest Fire (FF) network model [53] is similar to the BA model in the sense of network growth and preferential attachment that leads to networks with heavy-tailed degree distributions. However, the FF model has other properties such as densification power-law and shrinking diameter [53], which means that the network becomes denser and its diameter decreases as the network grows. The model starts with a single node and a new node v is added to the network as follows:

1. Uniformly select an existing node w and add edge (v, w).

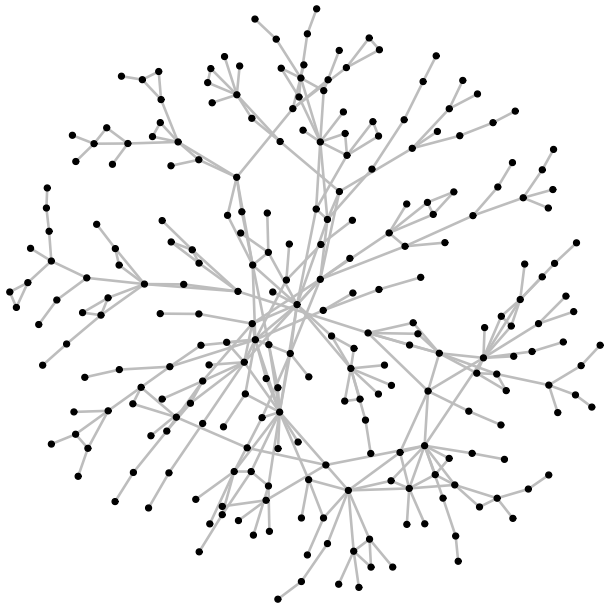
2. Randomly generate two numbers x and y that are binomially distributed with means

$$\frac{p}{1-p} \quad and \quad \frac{rp}{1-rp}, \tag{4.4}$$

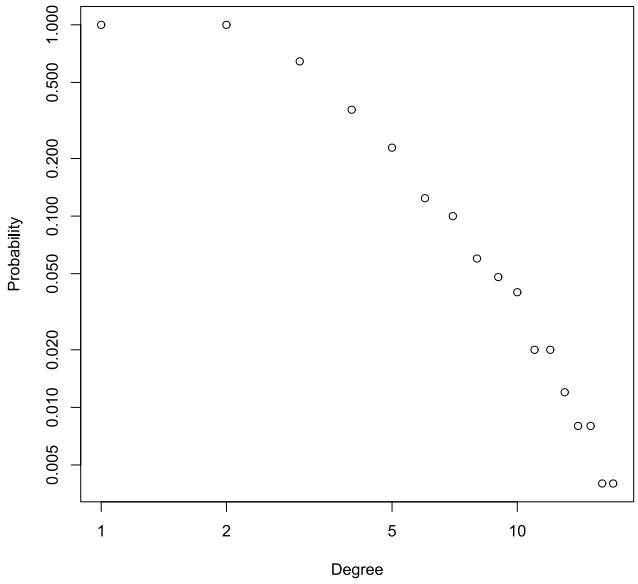
   respectively, where $0 < p < 1$ is called the forward probability, and $0 < r < 1$ is called the backward factor. Then, select x out-links and y in-links of w, and add an edge between v and each of those $(x+y)$ nodes. If node w did not have enough out-links or in-links, v connects to as many nodes as possible.

3. Apply step 2 recursively for all $(x+y)$ neighbours of node w. In order to prevent the process from cycling, each node is visited only once during the process.

The process is like spreading fire in a forest, since the burning starts with w and spreads to its $(x+y)$ neighbours and proceeds recursively for each of those nodes until it diminishes. The FF model was proposed in order to model some real networks such as autonomous systems, patents citations, and affiliation graphs, where they all have the shrinking diameter and densification power law properties [53]. An example FF network with $p = 0.25$, $r = 0.2$, and 250 nodes is shown in Figure 4.6, along with its degree distribution. Leskovec et al. [53] noted that the FF model generates networks with communities. Later, Leskovec et al. [54] used the network community profile plot to quantify the goodness of communities (the difference between intra-edges and inter-edges of communities decreases) in the FF networks, and their results showed that there exist communities in the tested FF networks, but as the size of these communities increases the quality of their goodness decreases.



(a) An example FF network with $p = 0.25$, $r = 0.2$, and 250 nodes.



(b) The log-log degree distribution of a FF network with 250 nodes.

Figure 4.6: An example FF network and its log-log degree distribution.

4.2 Weight Tuning

As mentioned in Section 3.2.1, the weights w_1 through w_4 in *DFSH* and the threshold of *DFSH-post* procedure should be determined for each tested network model and k-value. The purpose of tuning the undetermined weights and threshold is to perform *DFSH* on the unseen networks with similar topologies to the tested network models. The procedure of generating appropriate network samples and determining the best sets of weights for each network model and k-value is described in Section 3.2.1. The benchmark suite for each network model contains 2000 networks of different sizes ranging from 100 to 25,000 except for the benchmark suite of the ER model, which contains 420 networks of sizes ranging from 100 to 5000. The ER networks generated with small p values still contain too many edges (an ER network of size 5000 and $p = 0.001$ contains about 70,000 edges), and consequently a considerable amount of time is needed to do the weight tuning procedure on all 2000 ER networks since the runtime of the proposed heuristic is dependant on the number of edges of the network. Hence, the benchmark suite of the ER model contains networks of sizes up to 5000. The procedure of generating networks in BA and WS network models caused some node properties (such as cut vertex, local bridge, and bridge) used in *DFSH* to act neutral in ranking the nodes.

In the BA network model with $m = 1$, where each new node added to the network is only connected to one existing node, the generated network is a tree and any edge in the network is a bridge (no local bridge exists in a tree). Therefore, the weight w_2 of local bridges in *DFSH* is not needed to be determined and only the set of weights (w_1, w_3, w_4) should be determined. The BA networks generated by $m = 2$ (BA-m2) do not contain any bridges since each new node is added to two existing nodes, the procedure of generating the network started with two nodes connected to each other. Consequently, the weight w_3 of bridges was not considered during the experiments of calculating the best sets of weights (w_1, w_2, w_4). The WS network samples were generated by $k = 4$ and $p = 0.05$, and the calculated edge connectivity and vertex connectivity values for all network samples were at least 2, which means that no cut vertex or bridge exists in the generated network samples. Hence, the weights w_3 and w_4 of bridges and cut vertices, respectively, were not considered during the experiments on the WS networks. The best set of weights of *DFSH* for each tested network model and k-value, which were calculated based on the procedure given in Section 3.2.1, are presented in Table 4.1.

In order to show examples of distributions of the objective values across different combinations of weights, a network sample of size 5000 is selected for each network model, and the distribution of the objective values on weights are plotted when $k = 1\%$, except for the WS network sample that $k = 20\%$ was used since the objective values resulted by any tested weights were the same for $k < 20\%$. The tested values for each weight are in the range $[0, 0.1, 0.25, \ldots, 1]$ and the plots of all network models are shown in Figure 4.7.

	BA-m1			BA-m2			WS	
k-value	w_1	w_3	w_4	w_1	w_2	w_4	w_1	w_2
1%	0.1	0	0.85	0	0.7	0.1	0.25	0
10%	0.1	0.7	0.1	0	0.7	0.1	0.55	0
20%	0.25	0.1	0.4	0	0.4	0.25	0.7	0.25
30%	0	0.25	0.1	0	0.1	0.25	0.55	0.4
40%	0	0.25	0.1	0	0.85	0.1	0.25	0.85
50%	0	0.25	0.1	0.1	0.85	0.1	0.25	0.4
	ER				FF			
	w_1	w_2	w_3	w_4	w_1	w_2	w_3	w_4
1%	0.1	0.1	0.85	0.7	0.1	0.25	0.1	0.7
10%	0.1	0.25	0.4	0.4	0.1	0.4	1.15	2.8
20%	0	0.1	0.7	0.85	0.1	0.1	1.75	2.5
30%	0.1	0.85	0.85	0.55	0.1	0.1	2.8	2.8
40%	0.1	0.4	0.25	0.25	0.1	0.1	2.8	2.95
50%	0.25	0.55	0.55	0.1	0	0.25	0.85	0.4

Table 4.1: The best set of weights per each k-value for all tested network models. The size of benchmarks ranges from 100 to 25,000 in all network models except for the ER model, where the size of benchmarks ranges from 100 to 5000.

As shown in Figure 4.7(a), the objective values resulted by most of the weights in the FF network sample are around the minimum value found in tested weights, while a few sets of weights caused about 5 times larger objective values than the minimum. In the ER network sample, most of the sets of weights result in objective values near the maximum objective value, and a few sets of weights (37 out of 4096) result in objective values near the minimum value (Figure 4.7(b)). However, the difference between the maximum and minimum objective values resulted by weights in the ER network sample was lower than that observed in the FF network sample. As shown in Figures 4.7(c) and 4.7(d), the distribution of the objective values across all sets of weights shows variance objective values in the two BA network samples generated by $m = 1$ and $m = 2$. As shown in Figure 4.7(e), different objective values are resulted by tested sets of weights for the WS network, where most of the weights result in the highest objective value among other tested weights. However, the difference between the minimum and maximum objective values in the WS network sample is small, the maximum value is only 1.001 times bigger than the minimum value.

In order to show examples of the sensitivity of the calculated sets of weights in the tested network models, a network sample of size 5000 was selected from each benchmark suite and the distribution of objective values over small changes on calculated sets of weights when $k = 1\%$, except for the WS network where $k = 20\%$ was used, are presented in Figure 4.8. The weights are changed with 0.05 steps (the step of the tested weights in weight tuning experiments was 0.15). As can be observed from Figures 4.8(a) through 4.8(e), the objective values resulted by changed weights in all networks were in

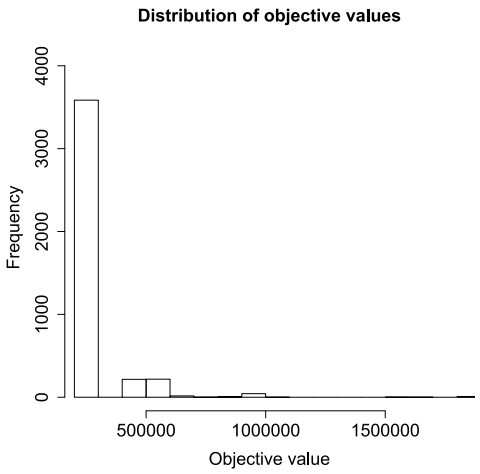
the range of values for all sets of weights shown in Figures 4.7(a) through 4.7(e). Moreover, the variations on weights did not cause considerable change in the objective value resulted by the calculated sets of weights that were shown in Table 4.1. For example, the calculated set of weights for the BA-m2 network when $k = 1\%$ results in objective value 10,136,316 and the best objective value gained by making small changes on weights is 10,127,313. Based on comparison between the plots represented in Figures 4.7 and 4.8, it can be concluded that the calculated sets of weights are not very sensitive to small changes, not much improvement was observed after making small changes on weights.

After the weights of *DFSH* were obtained for each k-value, extra experiments with the same procedure as stated in Section 3.2.1 were performed to calculate the threshold of the post-processing procedure *DFSH-post*. The tested thresholds were in the range $[0.1, 1]$ with a 0.05 step. The calculated thresholds for *DFSH-post* procedure per each k-value are given in Table 4.2 for all tested network models.

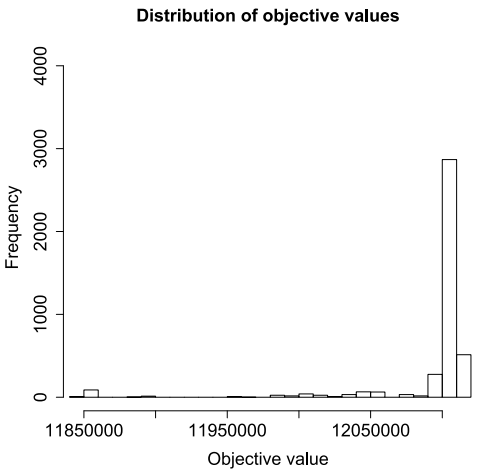
The objective values corresponding to different choices in threshold value in a network sample of size 5000 for each tested network model when $k = 20\%$ are plotted and shown in Figure 4.9. The plots show that different θ values may lead to different objective values, and therefore selecting a proper θ value is important in order to gain better performance from *DFSH-post*. As can be observed from Table 4.2, the calculated θ value for each network model increases for higher k-values in most of the cases, e.g., the threshold is $\theta = 1$ when $k = 50\%$ in the FF networks, while it is $\theta = 0.5$ when $k = 1\%$.

k-value	BA-m1	BA-m2	ER	WS	FF
1%	0.4	0.4	0.4	0.4	0.5
10%	0.5	0.5	0.45	0.8	0.6
20%	0.55	0.6	0.4	0.85	0.65
30%	0.55	0.65	0.4	0.85	0.65
40%	0.55	0.8	0.55	0.7	0.7
50%	0.55	0.7	0.55	0.7	1

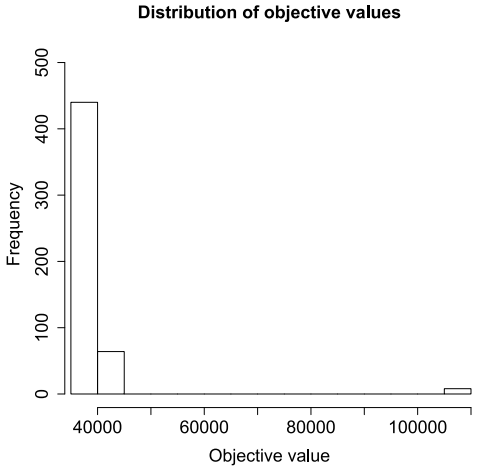
Table 4.2: The calculated threshold values per each k-value for all the tested network models.



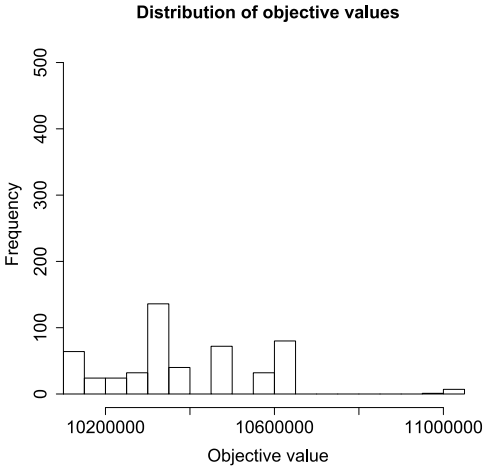
(a) Forest Fire



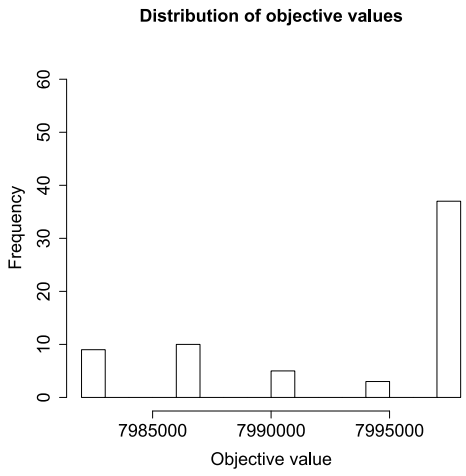
(b) Erdös-Renyi



(c) Barabasi-Albert, $m = 1$

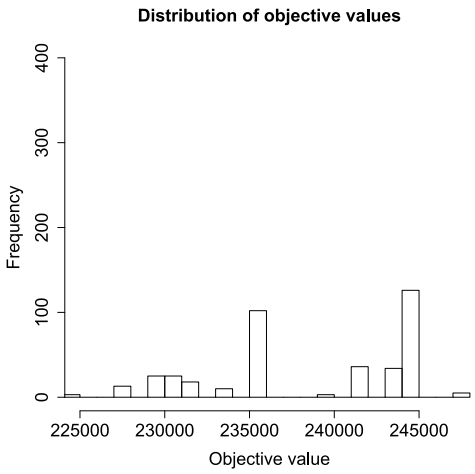


(d) Barabasi-Albert, $m = 2$

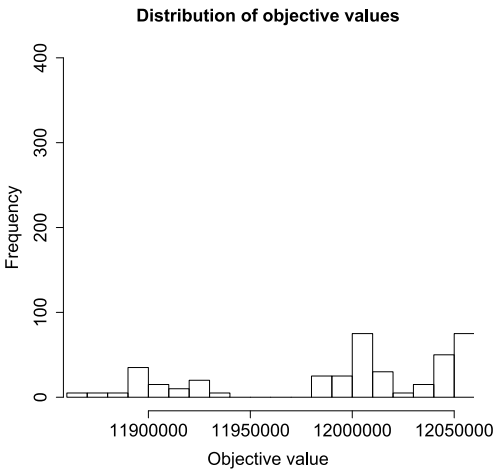


(e) Watts-Strogatz

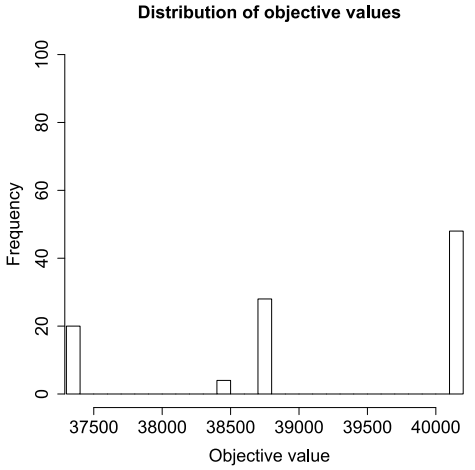
Figure 4.7: The distribution of objective values across all weights are plotted for a network sample of size 5000 for each tested network model when $k = 1\%$, except for the WS network where $k = 20\%$ was used.



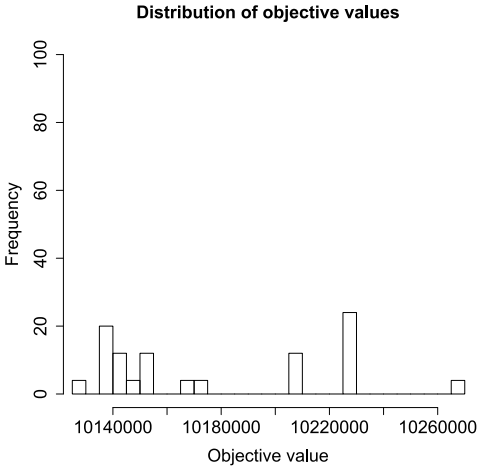
(a) Forest Fire



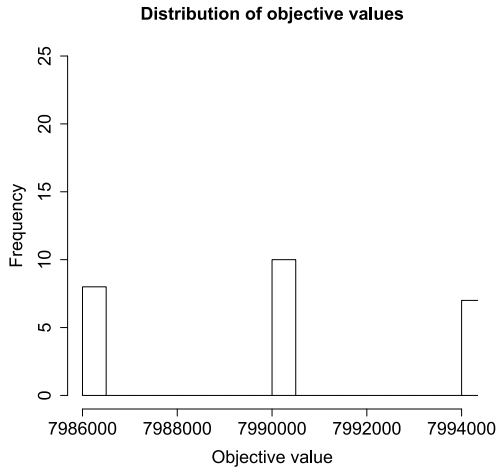
(b) Erdös-Renyi



(c) Barabasi-Albert, $m = 1$



(d) Barabasi-Albert, $m = 2$



(e) Watts-Strogatz

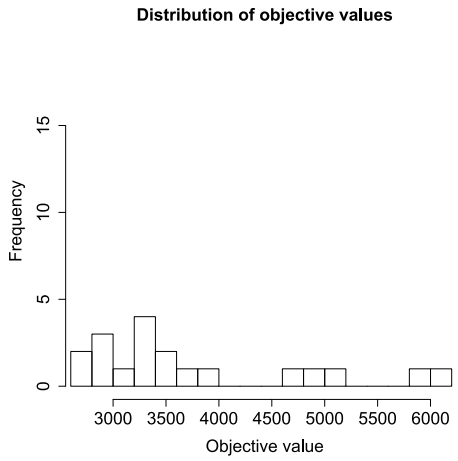
Figure 4.8: The distribution of objective values across weights with smaller steps are plotted for a network sample of size 5000 for each tested network model when $k = 1\%$, except for the WS network where $k = 20\%$ was used.

In order to show examples of the sensitivity of calculated thresholds, a network sample of size 5000 was selected from each network model, and then the objective values over the thresholds with smaller steps near the calculated θ when $k = 20\%$ is plotted. As an example, the threshold in the FF networks is $\theta = 0.65$ when $k = 20\%$, and the sensitivity is investigated by plotting the objective values resulted by thresholds in the range $[0.61, 0.69]$ with a 0.01 step, the 0.6 and 0.7 values were already considered in the main tuning procedure. The plots of tested network samples are shown in Figure 4.10. The plots represent the sensitivity of *DFSH-post* against small changes on θ for all network samples except in the WS network sample. The objective values resulted by new thresholds are in the range of the objective values calculated by all tested thresholds shown in Figure 4.9, and also the difference between the minimum and maximum objective values in changed thresholds are considerably smaller than what is observed for thresholds in the range $[0.1, 1]$. It can be concluded that the θ value is not sensitive to small changes in the tested network samples, although slightly better objective values may be achieved by tuning the θ with smaller steps as shown in Figure 4.10. In other words, the 0.05 steps used to calculate the best threshold value were sufficient to utilize the post-processor for improving the quality of solution of *DFSH*.

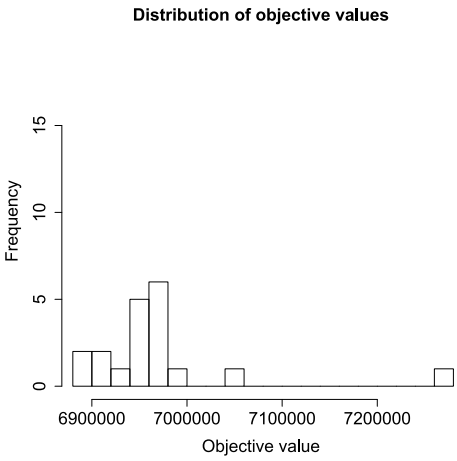
After the necessary weights are tuned, the objective values resulted by the proposed heuristic with post-processing procedure (*DFSH-post*) are compared to other centrality measures in order to evaluate the approaches on different network models with different properties. The results of comparisons on generated benchmark suites of the tested network models and further discussions are given in the following sections.

4.3 Small to Larger Size Networks

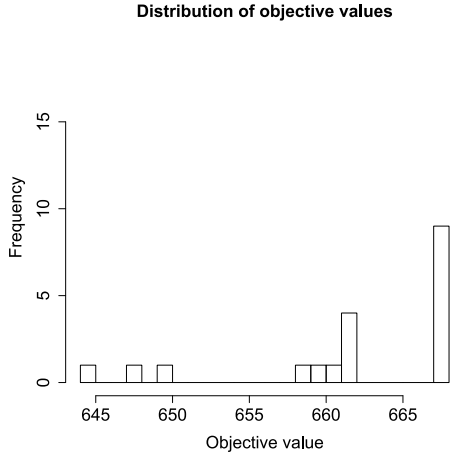
The network models introduced in this chapter are used to produce benchmark networks in order to evaluate the performance of the proposed heuristic and the centrality measures. Statistical comparisons help to provide evidence that the difference between approaches in not due to randomness.



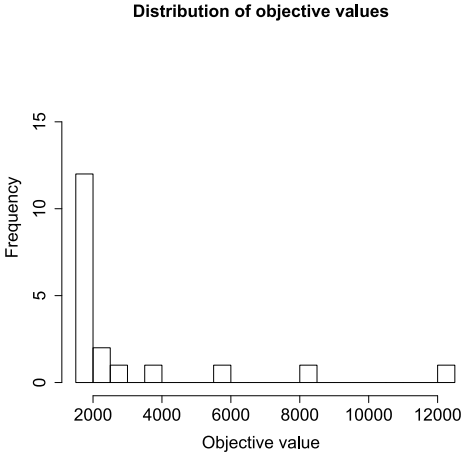
(a) Forest Fire



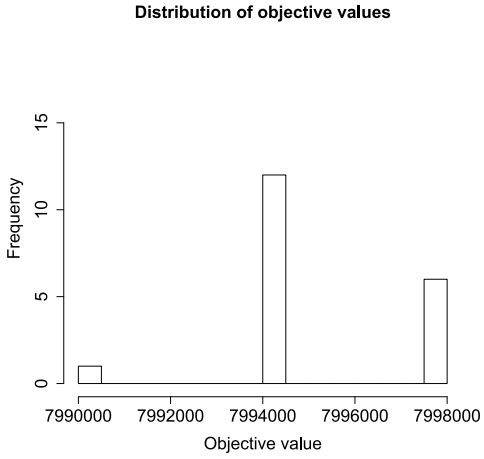
(b) Erdös-Renyi



(c) Barabasi-Albert, $m = 1$

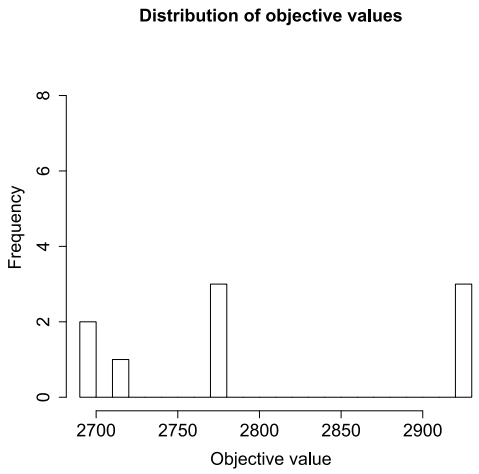


(d) Barabasi-Albert, $m = 2$

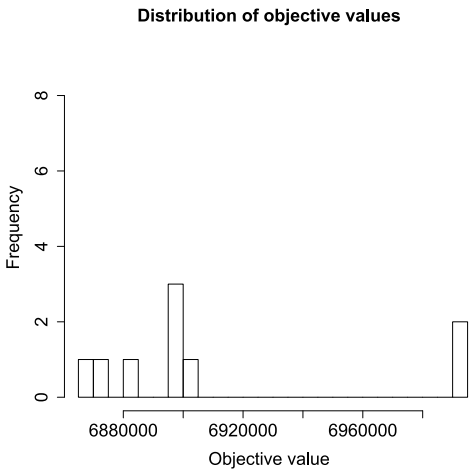


(e) Watts-Strogatz

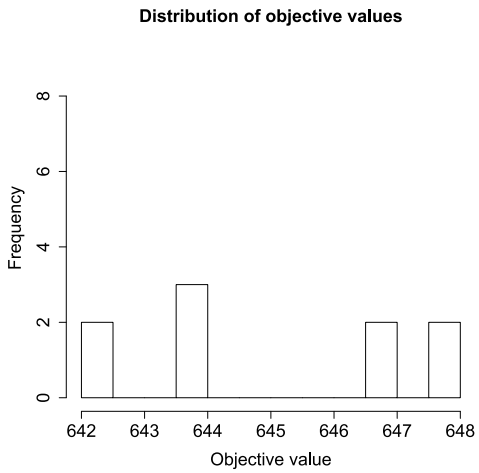
Figure 4.9: The distribution of objective values across all θ values of *DFSH-post* are plotted for a network sample of size 5000 for each tested network model when $k = 20\%$.



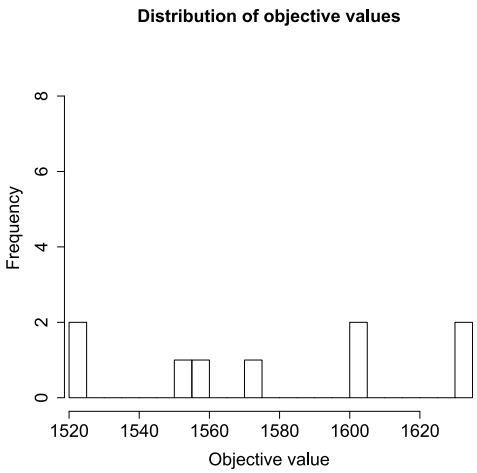
(a) Forest Fire



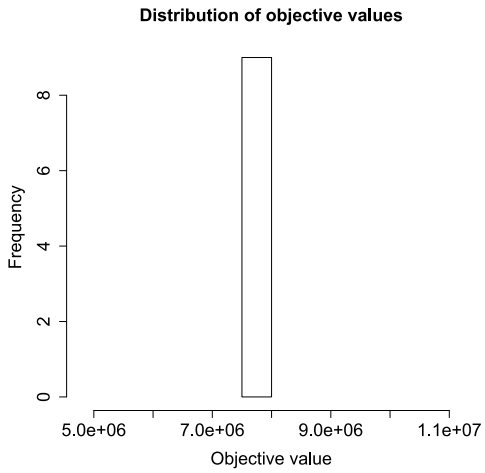
(b) Erdös-Renyi



(c) Barabasi-Albert, $m = 1$



(d) Barabasi-Albert, $m = 2$



(e) Watts-Strogatz

Figure 4.10: For analysing the sensitivity of calculated θ values to small changes, the distribution of objective values across θ values with smaller steps are plotted for a network of size 5000 per each tested network model when $k = 20\%$.

4.3.1 Benchmark Networks

For each network model introduced in Section 4.1 a benchmark suite containing 2000 networks of different sizes ranging from 100 to 25,000 was generated, except for the ER model, which contains 420 networks. The benchmark networks were generated with the same procedure as discussed in Section 3.2.1. The number of vertices, edges, average degree, and clustering coefficient of a few network sizes are shown in Tables 4.3 through 4.7 for BA-m1, BA-m2, WS, FF and ER network models, respectively. There are a total number of 100 network sizes used for each network model except for the ER model, which 21 network sizes were used. The clustering coefficient of any BA-m1 network is zero due to the fact that no cycle and consequently no triangle exists in a tree.

n	m
1000	999
2500	2499
5000	4999
7500	7499
10,000	9999
12,500	12,499
15,000	14,999
20,000	19,999
24,000	23,999

Table 4.3: The number of vertices n and edges m of nine BA-m1 network sizes

n	m	$\langle k \rangle$	C
1000	1998	3.99	0.008
2500	4998	3.99	0.004
5000	9998	3.99	0.002
7500	14,998	3.99	0.001
10,000	19,998	3.99	0.001
12,500	24,998	3.99	0.0009
15,000	29,998	3.99	0.0009
20,000	39,998	3.99	0.0007
24,000	47,998	3.99	0.0006

Table 4.4: The number of vertices n, edges m, average degree $\langle k \rangle$, and clustering coefficient C of nine BA-m2 network sizes

n	m	$\langle k \rangle$	C
1000	4000	8	0.473
2500	10,000	8	0.465
5000	20,000	8	0.468
7500	30,000	8	0.466
10,000	40,000	8	0.466
12,500	50,000	8	0.466
15,000	60,000	8	0.465
20,000	80,000	8	0.466
24,000	96,000	8	0.466

Table 4.5: The number of vertices n, edges m, average degree $\langle k \rangle$, and clustering coefficient C of nine WS network sizes

n	m	$\langle k \rangle$	C
1000	1407	2.81	0.214
2500	3513	2.81	0.212
5000	7008	2.80	0.202
7500	10,552	2.81	0.200
10,000	14,095	2.81	0.198
12,500	17,573	2.81	0.196
15,000	21,094	2.81	0.195
20,000	28,157	2.81	0.192
24,000	33,755	2.81	0.192

Table 4.6: The number of vertices n, edges m, average degree $\langle k \rangle$, and clustering coefficient C of nine FF network sizes

n	m	$\langle k \rangle$	C
1000	9979	19.95	0.019
2000	19,235	19.23	0.009
3000	21,142	14.09	0.004
4000	54,375	27.18	0.006
5000	70,049	28.01	0.005

Table 4.7: The number of vertices n, edges m, average degree $\langle k \rangle$, and clustering coefficient C of five ER network sizes

4.3.2 Experimental Results and Discussions

A strategy is needed to compare the quality of solution of two different approaches for generated benchmark data and determine which approach outperform the other one. The following steps describe the strategy of how any two approaches are compared for a benchmark suite in order to determine whether the difference between their performances is due to randomness. As described above, 20 network samples were generated for each network size of any benchmark suite generated in this thesis, since the experiments need a considerable amount of time if 30 samples for each network size are generated.

1. Calculate the objective values resulting from the two approaches we want to compare to each other for each network sample after deleting k selected nodes, where the k-value is in the range of 1% to 50% of the network's size with a 10% step.

2. For each network size with 20 network samples of the same size do:

   (a) Calculate the number of network samples in which each approach has lower objective value than the other one, which are represented as a_1 and a_2 for approaches F_1 and F_2, respectively. The number of samples that the approaches result in equal objective values is considered as the number of ties between them, which is shown as t.

   (b) We need to determine if an approach with higher number of wins (calculated in previous step) is significantly better than the other one. For example, if $a_1 = 12$, $a_2 = 5$, and $t = 3$, F_1 may not be significantly better than F_2 for this network size. The binomial test is used in this case. So, perform a binomial test (described in Appendix B) between the approaches with considering $(a_1 + t)$ and $(a_2 + t)$ the number of wins of the F_1 and F_2 approaches, respectively. If the p-value of binomial test is less than 0.05, the approach with the higher number of wins will receive 1 score, the scores are represented as A_1 and A_2 for the approaches F_1 and F_2, respectively. The tie score T is increased by 1 when the p-value of binomial test is greater than 0.05, since $p > 0.05$ indicates that the difference between the two tested approaches is not significant.

3. After the A_1, A_2, and T scores are calculated for all 100 network sizes (21 network sizes for the ER model) perform a binomial test between the approaches with considering $(A_1 + T)$ and $(A_2 + T)$ the number of wins of the F_1 and F_2 approaches, respectively. The highest possible value for A_1, A_2, or T is 100, since the total number of network sizes in each generated benchmark suite is 100, except for the ER benchmark data that contains 21 network sizes. Therefore, the highest value for the A_1, A_2, or T is 21 for the benchmark suite of the ER model. If the p-value

of binomial test is less than 0.05, the approach with higher score is reported as the winner, otherwise both approaches are considered to have equivalent performance for the tested benchmark suite and k-value. For example, $A_1 = 85$ means that the approach F_1 wins at 85 cases out of 100, and if $A_2 = 10$ and $T = 5$, the p-value of the binomial test is near zero, and consequently F_1 is significantly better than F_2 for the tested benchmark data and k-value.

The connections between the nodes of a network are different than other generated network samples of the same size, and the objective value of a network is related to the connections between the nodes of the network and not only the number of edges. That is, the objective value resulted by the optimal solution may be different for two network samples of the same size and number of edges. Therefore, it is not applicable to use the t-test to compare two approaches based on the objective values due to the fact that the objective value of each network sample is independent from other network samples of the same size. A solution for this problem is to use the binomial test (described in Appendix B), which compares the number of times each approach has better objective value than the other one in tested networks (named as number of wins) in order to determine whether the approach with higher number of wins is significantly better than another (significant at the 5% level).

Comparing *DFSH* and *DFSH-post*

DFSH and *DFSH-post* approaches are compared in all generated benchmark suites. The binomial test results are presented in Tables B.1 through B.5 with the number of wins of each approach and the calculated p value for any tested k-value of each benchmark suite, and they are given in Appendix B. The results of experiments indicated that *DFSH-post* has either the same performance as *DFSH* or better than *DFSH* in all tested benchmark networks. The effect on the objective value after removing $k = 20\%$ of vertices by DFSH and DFSH-post procedures is shown in Figure 4.11 for each of the five benchmark networks when $k = 20\%$.

As shown in Figure 4.11, *DFSH-post* did not worsen the quality of solution of *DFSH* in any benchmark data, and the improvement in objective value is observable for benchmark suites regarding the FF, BA-m1, and BA-m2 networks. In WS networks, the difference between *DFSH* and *DFSH-post* is not clear in the plot shown in Figure 4.11(e) because the objective values of larger networks are way higher than networks of smaller sizes, which made it hard to show the difference between approaches in diagram. However, *DFSH-post* slightly improved the results of *DFSH* in almost all of the WS network samples, e.g., the effect on the objective value after removing $k = 20\%$ of vertices for a WS network sample of size 3500 by *DFSH* and *DFSH-post* procedures is 3,907,410 and 3,901,822, respectively. Both approaches performed similarly on the ER network samples because the connection between nodes in the ER network is uniformly at random,

which results in networks with no power-law degree distribution or community structure (see Section 4.1.1).

Comparing *DFSH-post* and Centrality Based Approaches

The experiments showed that the performance of *DFSH-post* is as good as or better than *DFSH* in the tested benchmark suites, and therefore *DFSH-post* is selected to be compared to the centrality based approaches proposed in Section 2.5. The procedure of scoring approaches presented in this chapter (based on using binomial test) was used to compare *DFSH-post* to other centrality based approaches. The detailed results of binomial tests between each two centrality measures and the results of binomial tests between the best centrality measure and *DFSH-post* per k-value for all tested benchmark suites are given in Appendix B. Table 4.8 shows the approach that was significantly better than other tested approaches in each network model, and Table 4.9 shows the number of times each approach was declared as the best in Table 4.8. The purpose of Table 4.9 is to compare the performance of approaches in overall for all tested benchmark data.

k	FF	BA-m1	BA-m2	ER	WS
1%	Deg & *DFSH-post*	Between & *DFSH-post*	Deg & Page & *DFSH-post*	all	all
10%	Page & *DFSH-post*	*DFSH-post*	Page & *DFSH-post*	all	Close
20%	*DFSH-post*	*DFSH-post*	*DFSH-post*	all	Close
30%	*DFSH-post*	Page	Page	all	Deg
40%	Page	all except Close	Page & *DFSH-post*	all	*DFSH-post*
50%	*DFSH-post*	all except Close	Page	all	*DFSH-post*

Table 4.8: The winner of comparisons between *DFSH-post* and other centrality based approaches

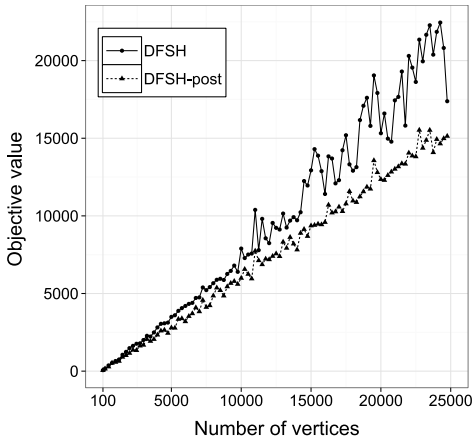
DFSH-post	Closeness	Betweenness	Degree	PageRank
23	9	10	12	17

Table 4.9: The number of times each approach was declared as the winner for small to larger size benchmark networks

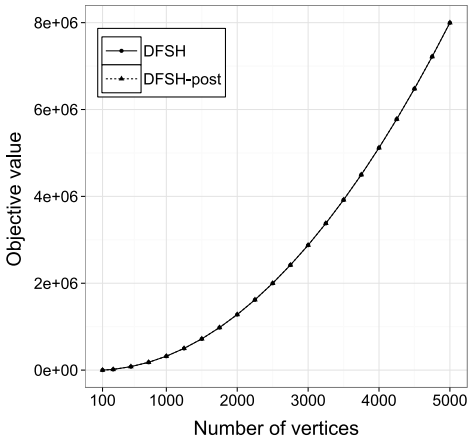
The results of experiments on the BA-m1 network samples show that the closeness centrality had the worst objective value in all tested k-values. The closeness ranks any node v in the graph based on its average shortest path to other vertices, which results in ranking the neighbours of the central nodes of the input graph (a central node is a node that its distance to the rest of the graph is the lowest) higher than some other nodes that were more important regarding the objective of the CNDP. For example, the objective values of all other approaches are zero for all tested BA-m1 networks when $k > 30\%$, while the closeness even selected nodes of degree 1 since they were neighbours of central nodes, none of the other approaches rank nodes of degree 1 higher than other nodes.

The betweenness showed similar performance as *DFSH-post* when $k = 1\%$ since it ranks the nodes responsible for the interconnectedness of a BA network (the nodes connecting different parts of a graph) higher than others [24], and for $k = 1\%$ it was sufficient to remove those kind of nodes that result in lower objective value than other centrality based approaches. However, the betweenness was not able to result in a lower objective value compared to other approaches for $10\% \leq k \leq 30\%$ due to the fact that the nodes selected by betweenness are connected to each other and selecting some of them are not necessary since most of their neighbours are already selected, the same problem arose for *DFSH* that was the reason to design *DFSH-post* to solve it (see Section 3.3). It was shown in previous works that the scale-free networks are vulnerable when attacking the nodes of highest degrees [2, 23, 24], but the results of these comparisons showed that even lower objective values were produced by *DFSH-post* or PageRank, both of these approaches select nodes that are not necessarily neighbours in set L.

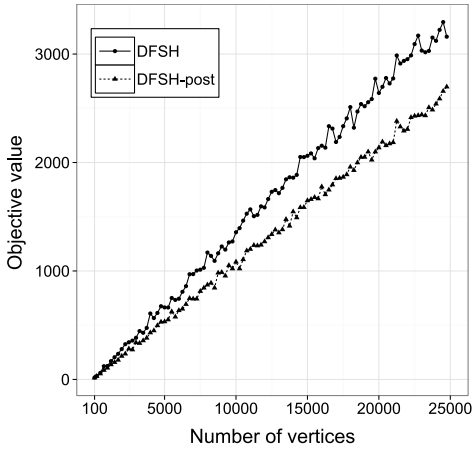
The effect on the objective value after removing different k number of nodes by tested approaches is shown in Figure 4.12 for BA-m1 benchmark networks. The closeness centrality results in considerably higher objective values than other approaches in BA-m1 networks (examples given in Table B.27), and the difference between the other tested approaches is not observable in the diagram any more for $k < 40\%$ if the closeness is plotted as well. Therefore, the closeness is not presented for $k < 40\%$ cases. As mentioned earlier, the effect on the objective value after removing $k \geq 40\%$ of vertices by all approaches, except for the closeness, is equal to zero for BA-m1 networks. The closeness is presented in Figures 4.12(e) and 4.12(f), where other approaches result in objective value equal to zero for all the network samples. The PageRank performed better than other approaches when $k = 30\%$ as shown in Figure 4.12(d), and *DFSH-post* is the second best. However, the difference between approaches in larger network sizes is around 300, which is a small number considering the large network size. For example, the highest possible objective value after removing $k = 30\%$ of vertices for a network with 25,000 vertices is 28,121,250, and thus the difference of 300 between the approaches is considered negligible.



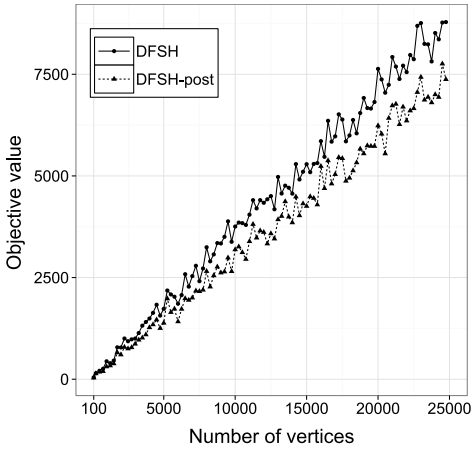
(a) Forest Fire



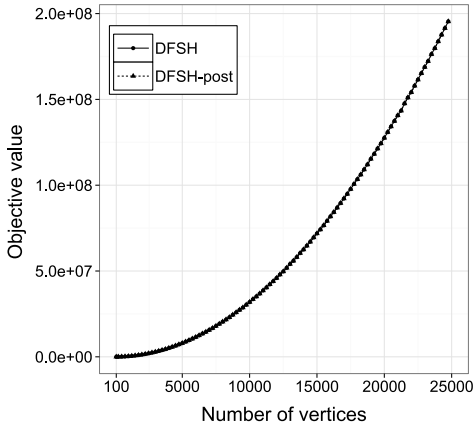
(b) Erdös-Renyi



(c) Barabasi-Albert, $m = 1$

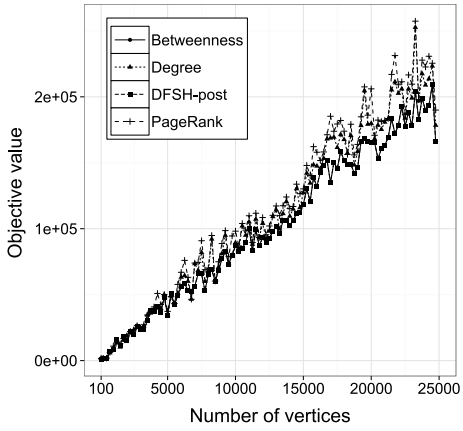


(d) Barabasi-Albert, $m = 2$

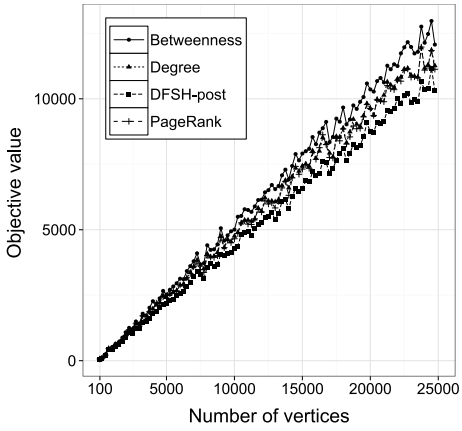


(e) Watts-Strogatz

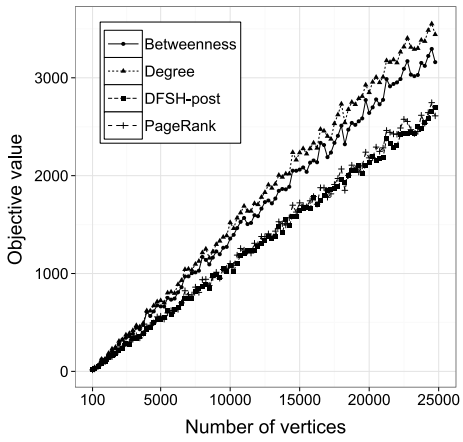
Figure 4.11: The effect on the objective value after removing $k = 20\%$ of vertices by DFSH and DFSH-post procedures for each of the five benchmark networks.



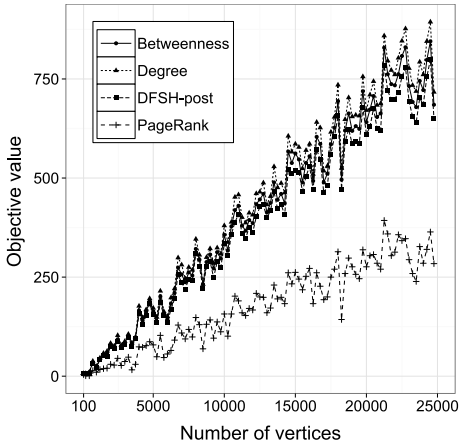
(a) $k = 1\%$ of vertices



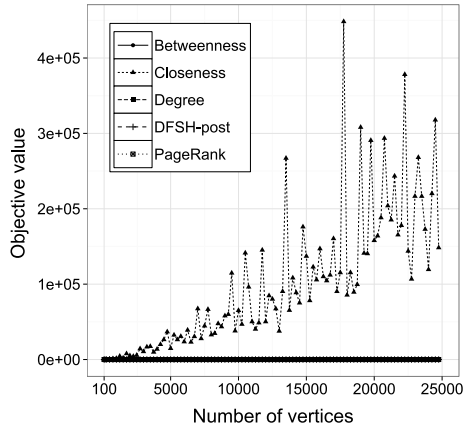
(b) $k = 10\%$ of vertices



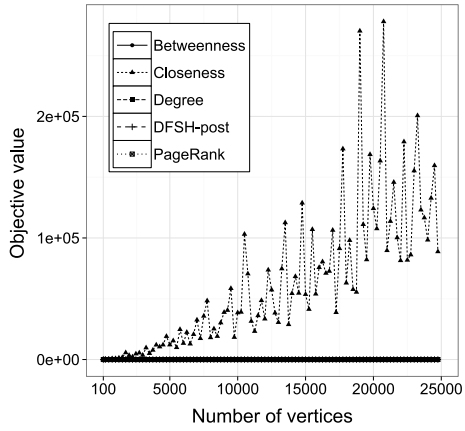
(c) $k = 20\%$ of vertices



(d) $k = 30\%$ of vertices



(e) $k = 40\%$ of vertices

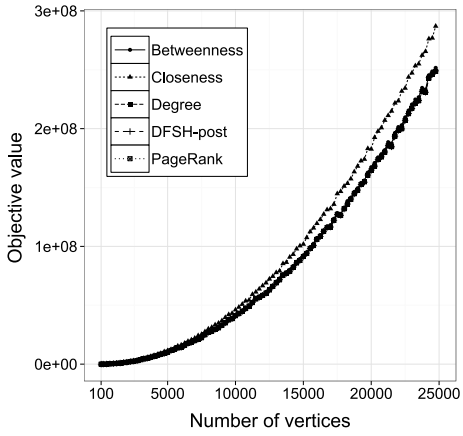


(f) $k = 50\%$ of vertices

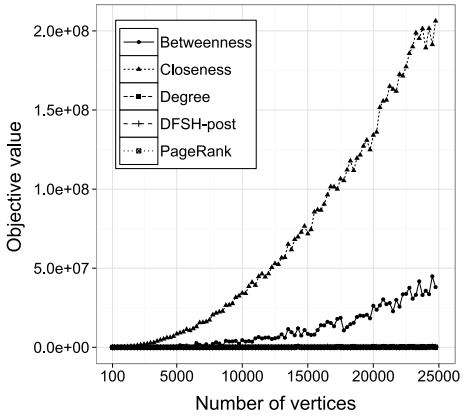
Figure 4.12: The effect on the objective value after removing different k number of vertices by tested approaches for BA-m1 network samples. The closeness is not plotted when $k < 40\%$ since the objective values resulted by that approach are higher than the results of other tested approaches.

It is harder for the tested approaches to solve the CNDP for the BA-m2 network samples than in the BA-m1 networks due to fact that no bridge exists in the BA-m2 networks (see Section 4.1.3). The closeness again has the worst performance compared to other approaches since it ranks the neighbours of central nodes higher than other nodes that are important regarding the objective of the CNDP, the new nodes added to the central nodes of the network are not yet part of any backbone of the network. In other words the closeness gives higher ranks to nodes that are unnecessary to be removed only because they are connected to the central nodes of the network. In BA-m2 networks, each new node is added to two existing nodes, and it leads to lower average path length D (Eq. (2.13)) in the BA-m2 networks than in the BA-m1 networks, and there are more paths between any pair of nodes in these networks than what is observed in BA-m1 networks. Therefore, it is harder for betweenness to rank the nodes with respect to the objective of the CNDP, and the objective values resulted by the betweenness indicated that this approach is not able to outperform other tested approaches for the BA-m2 networks. Both of the PageRank and *DFSH-post* approaches result in lower objective values than degree centrality in the BA-m2 networks since they are able to rank the nodes not only based on their degrees but also based on the the importance of their neighbours. It is obvious from the ranking procedure of the PageRank that it ranks any node based on the ranks of its neighbours (see Section 2.5.4), and *DFSH-post* ranks each node with considering the vertex similarity between that node and its neighbours.

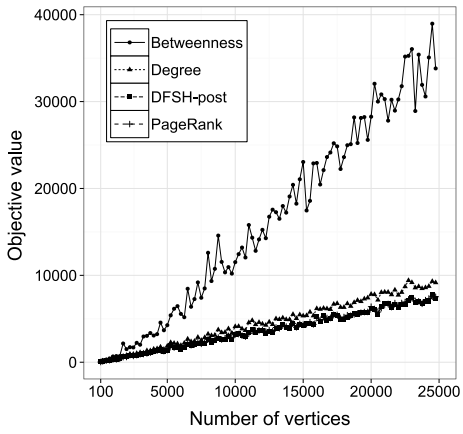
The effect on the objective value after removing different k number of nodes by tested approaches is shown in Figure 4.13 for BA-m2 benchmark networks. Since the closeness results in much larger objective value than other approaches and plotting its results prevent to see the difference between other approaches for $k \geq 20\%$, it was not shown in Figures 4.13(c) through 4.13(f). For example, the effect on the objective value after removing $k = 10\%$ of vertices for a BA-m2 network of size 1000 by closeness is about 250,000, while all other approaches result in objective value less than 10,000 (more examples are given in Table B.28). As can be observed from the plots, the degree centrality results in less objective values than betweenness since the BA-m2 networks are scale-free and the BA networks are vulnerable to the deletion of nodes of highest degrees as Albert et al. [2] stated. However, only selecting the nodes of highest degrees was not the best strategy as the results show that *DFSH-post* and PageRank lead to lower objective values than degree centrality. Although the binomial tests between the PageRank and *DFSH-post* approaches indicated that one of them is better than another for $k = 20\%$, 30%, and 50%, their effect on the objective value is close to each other compared to the effect on the objective value by closeness or betweenness as shown in Figures 4.13(a) through 4.13(f). Based on the results, *DFSH-post* is still among those approaches that result in lowest objective values for the BA-m2 networks, which are harder than the BA-m1 networks, while the effect on the objective value by betweenness is not among the best.



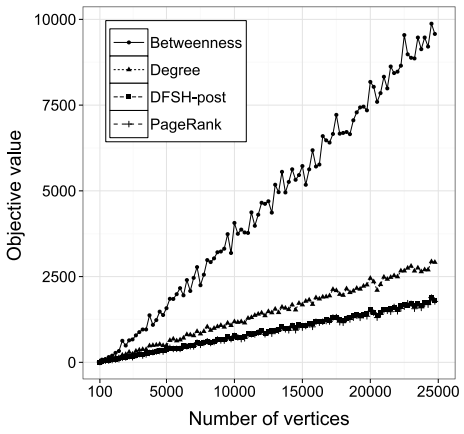
(a) $k = 1\%$ of vertices



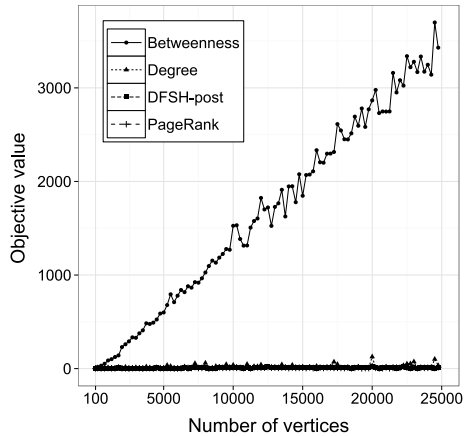
(b) $k = 10\%$ of vertices



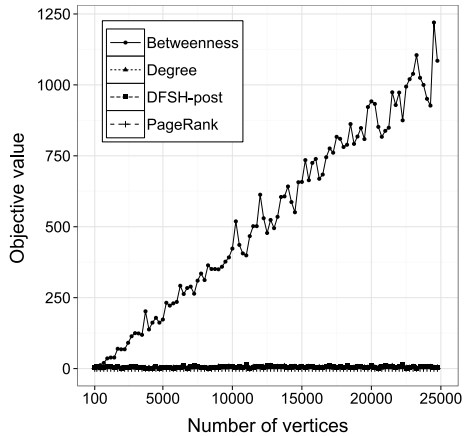
(c) $k = 20\%$ of vertices



(d) $k = 30\%$ of vertices



(e) $k = 40\%$ of vertices

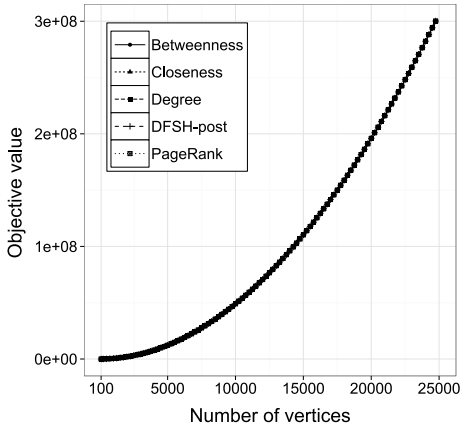


(f) $k = 50\%$ of vertices

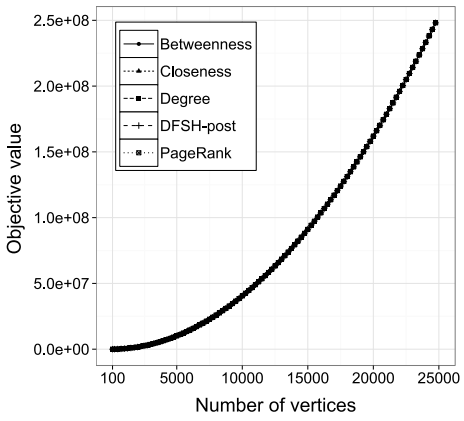
Figure 4.13: The effect on the objective value after removing different k number of vertices by tested approaches for BA-m2 network samples. The closeness is not plotted when $k > 10\%$ since the objective values resulted by that approach are higher than the results of other tested approaches.

The tested approaches were unable to limit the size of the largest component in the WS networks when $k = 1\%$ due to the high clustering of these networks and the lack of power-law degree distribution. The effect on the objective value after removing different k number of nodes by tested approaches is shown in Figure 4.14 for WS benchmark networks. As can be observed from Figures 4.14(b) and 4.14(c), although the closeness is determined as the winner for $k = 10\%, 20\%$ based on the results of binomial tests, the difference between the performance of the closeness and other approaches is not noticeable. The degree centrality results in lower objective values than other approaches when $k = 30\%$ for the WS networks, but still the performances of approaches are about the same as can be observed from Figure 4.14(d). In larger tested k-values ($k > 30\%$) the centrality based approaches select nodes that are neighbours of each other since there is no priority about not selecting nodes that most of their neighbours are already selected. However, this problem was solved by using the proposed post processing for *DFSH* heuristic. The difference between the effect on the objective value by centrality based approaches and *DFSH-post* can be easily observed in Figures 4.14(e) and 4.14(f), which is because of the problem of selecting nodes neighbours to each other that is solved by *DFSH-post*.

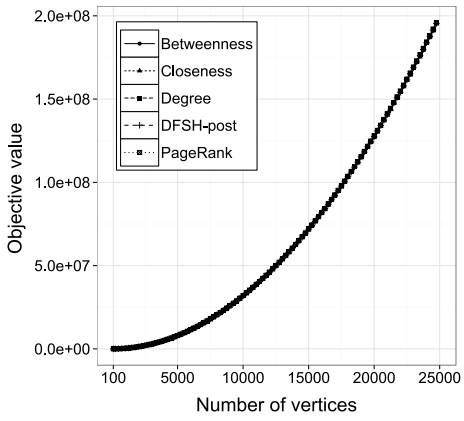
The generated ER network samples have many edges, although the p value of connecting each pair of nodes during generating ER networks was set to very small numbers (i.e., less than $p = 0.001$), examples of number of edges for five different ER network sizes are shown in Table 4.7. None of the tested approaches is able to outperform others based on the results of binomial tests (significant at the 5% level) for the ER network samples. The effect on the objective value after removing different k number of nodes by tested approaches is shown in Figure 4.15 for ER benchmark networks. The centrality measures and *DFSH-post* are unable to limit the size of the largest component of the ER networks as good as what observed for other benchmark data since the ER networks contain much more edges compared to other benchmark networks. For example, a generated ER network of size 5000 and $p = 0.001$ contains about 70,000 edges while the maximum number of edges for networks generated by other models with 5000 nodes is at most 20,000.



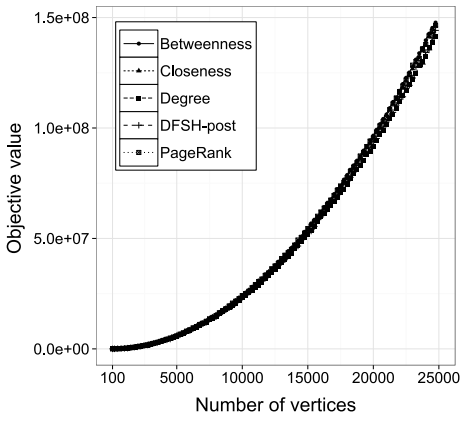
(a) $k = 1\%$ of vertices



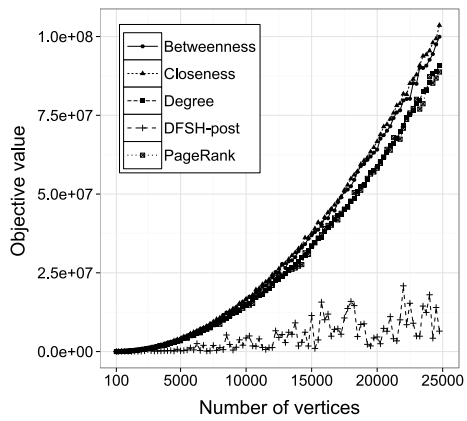
(b) $k = 10\%$ of vertices



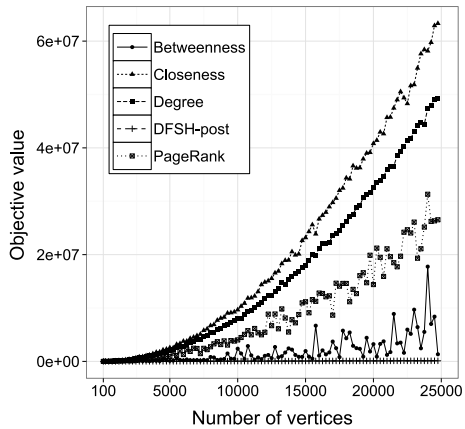
(c) $k = 20\%$ of vertices



(d) $k = 30\%$ of vertices

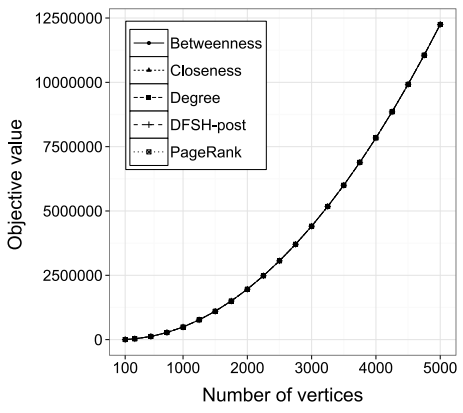


(e) $k = 40\%$ of vertices

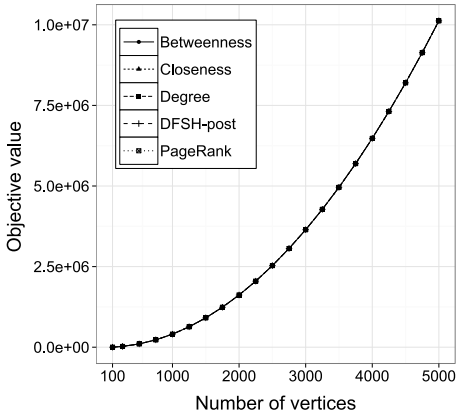


(f) $k = 50\%$ of vertices

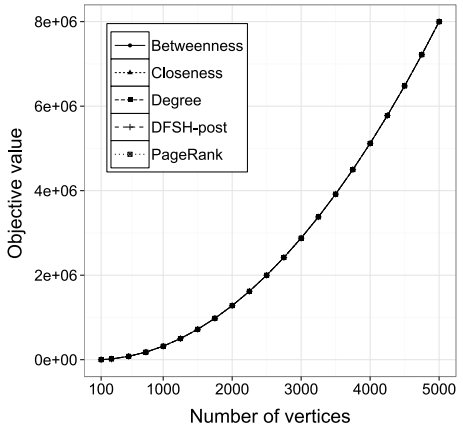
Figure 4.14: The effect on the objective value after removing different k number of vertices by tested approaches for WS network samples.



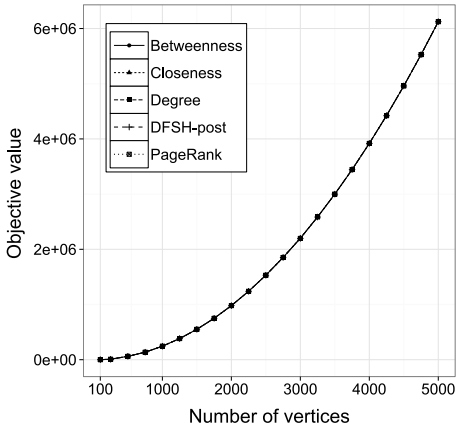
(a) $k = 1\%$ of vertices



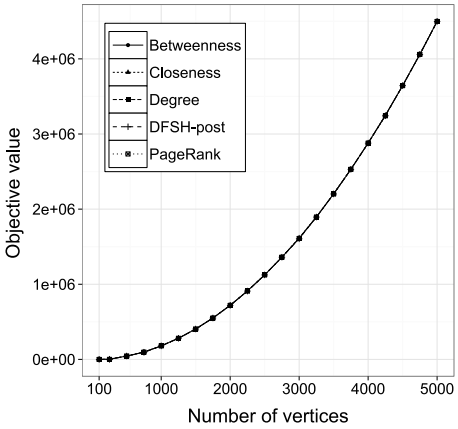
(b) $k = 10\%$ of vertices



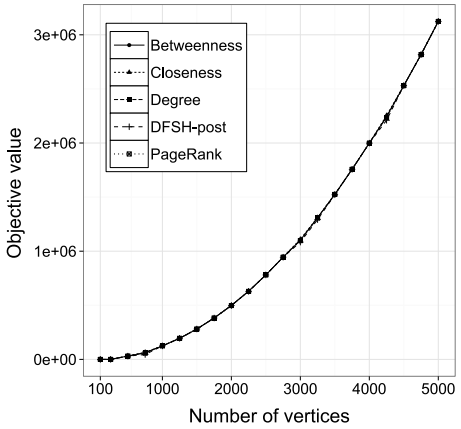
(c) $k = 20\%$ of vertices



(d) $k = 30\%$ of vertices



(e) $k = 40\%$ of vertices



(f) $k = 50\%$ of vertices

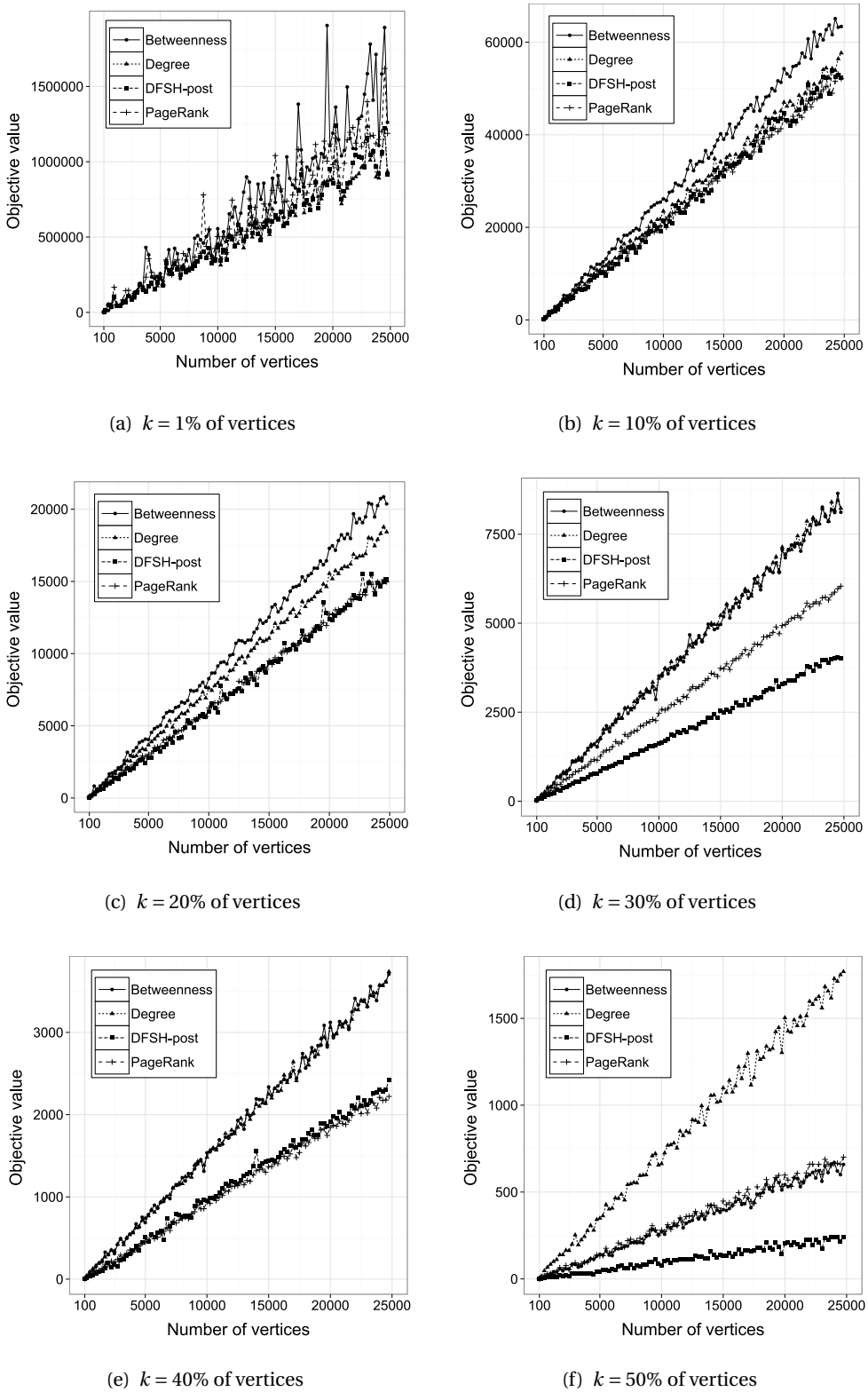
Figure 4.15: The effect on the objective value after removing different k number of vertices by tested approaches for ER network samples.

The network samples generated by the FF model have a tree-like structure similar to the BA model and also have a heavy tailed degree distribution because of preferential attachment during network growth. Similar to what observed in BA networks, the closeness centrality is not able to result in lower objective values than other approaches since it ranks the central nodes and their neighbours higher than other nodes that are more important with regard to the objective of the CNDP. The effect on the objective value after removing different k number of nodes by tested approaches is shown in Figure 4.16 for FF benchmark networks. The effects on the objective value by the closeness is not plotted because of the huge difference between its results and the results of the other approaches, e.g., the closeness results in objective value about 730,000 for a FF network of size 15,000 while the other approaches result in objective value less than 40,000 (more examples are given in Table B.26). The degree centrality results in lower objective than other approaches when $k = 1\%$ because of the heavy tailed degree distribution in the FF networks.

In overall, *DFSH-post* is competing with the best approaches in all tested k-values since the post processing procedure helps the heuristic to deselect unnecessary nodes (the nodes that had many neighbours in set L) and select new important nodes instead. In total 30 cases of benchmark data and k-values, *DFSH-post* results in the lowest objective value in 23 cases as shown in Table 4.8, the comparisons between approaches for each case are done based on binomial test (significant at the 5% level). As bolded in Table 4.9, *DFSH-post* has the highest number of wins among other approaches. The runtime and objective values resulted by tested approaches in different network sizes for each tested network model are given in Appendix B. The runtime of degree centrality, PageRank, and *DFSH-post* is less than 10 seconds in all tested networks, while closeness and betweenness need much more time to complete, and their runtime goes up to about 180 seconds. The degree centrality has the lowest runtime among other approaches.

Discussion

The comparison between *DFSH-post* and tested centrality measures shows that the quality of solution of *DFSH-post* is as good as or better than *DFSH* itself in all tested benchmark suites. Moreover, the results of comparisons between *DFSH-post* and tested centrality measures suggest that the proposed heuristic results in better objective value than the other approaches in many cases (23 out of 30). The runtime of the proposed heuristic is much less than closeness and betweenness while the objective value resulted by the heuristic is better than those approaches in many cases. Since the weights of *DFSH-post* are tuned for tested benchmark data, comparing its results to other approaches may not seem fair. In this regard, *DFSH-post* is compared to the centrality measures in unseen benchmarks in the next section.



(a) $k = 1\%$ of vertices

(b) $k = 10\%$ of vertices

(c) $k = 20\%$ of vertices

(d) $k = 30\%$ of vertices

(e) $k = 40\%$ of vertices

(f) $k = 50\%$ of vertices

Figure 4.16: The effect on the objective value after removing different k number of vertices by tested approaches for FF network samples. The closeness is not plotted since the objective values resulted by that approach are higher than the results of other tested approaches.

4.4 Small Size Networks

Ventresca [72] presented two population based approaches (SA and PBIL) for the CNDP. He compared the objective values resulted by those population based approaches to each other and to random deletion of nodes. In this section the results of the proposed heuristic are compared to the results of population based approaches in [72].

4.4.1 Benchmark Networks

The comparisons are made based on 16 different benchmark networks generated by the BA-m1, FF, ER, and WS models. The size of benchmark networks varies from 250 to 5000 nodes, which are considered as small networks. The number of vertices, edges, and critical nodes to be deleted (k-value) for each network sample is given in Table 4.10.

Problem	Vertices	Edges	k
FF250	250	400	50
FF500	500	792	110
FF1000	1000	1633	150
FF2000	2000	4046	200
ER235	235	349	50
ER465	465	699	80
ER940	940	1399	140
ER2343	2343	3499	200
BA500	500	499	50
BA1000	1000	999	75
BA2500	2500	2499	100
BA5000	5000	4999	150
WS250	250	1250	70
WS500	500	1500	125
WS1000	1000	5000	200
WS1500	1500	4500	265

Table 4.10: The number of edges and vertices of small networks and their related number of k-critical nodes

4.4.2 Experimental Results and Discussions

The SA and PBIL approaches were run for 30 trials, and the minimum, average, and maximum objective values for each benchmark instance are calculated [72]. In order to compare the results of the proposed heuristics and other centrality measures given in Section 2.5 to these population based approaches, the minimum objective value resulted by the SA and PBIL during 30 trials is used as their best results. The effect on the objective value by each approach are presented in Table 4.11, and the lowest objective value for each benchmark network is bolded in order to indicate the winner. The last row in Table 4.11 calculates the number of times each approach has the best result.

DFSH-post results in objective value as good or better than *DFSH* for all tested benchmarks. The ER benchmark instances generated in [72] had much less number of edges than what is used in the ER networks generated here, which consequently led to different results than in Section 4.3. As can be seen, the results of *DFSH* are similar to other centrality measures in the ER networks, where using the proposed post-processing procedure results in lower objective values than *DFSH*. Except for the closeness, the effect on the objective value by other centrality measures are close to each other, e.g., the objective values for BA500 are 240, 269, and 238 for the degree centrality, betweenness, and PageRank, respectively. The closeness results in higher objective values than other approaches in most of the cases (12 out of 16) since closeness ranks the central nodes and their neighbours higher than other nodes that are more important to be removed with regard to the objective of the CNDP. The results show that the effect on the objective value after removing nodes calculated by closeness is comparable to other centrality measures for the WS networks because of the lack of heavy tailed degree distribution and community structure. Similar to the results of binomial tests presented in Section 4.3, the *DFSH-post* results in objective values that are always among the best objective values, and also it results in the lowest objective value among other tested approaches for most of the benchmark problems (11 out of 16 problems).

Problem	SA	PBIL	*DFSH*	*DFSH-post*	Degree	Closeness	Betweenness	PageRank
ER235	7700	6700	5327	**1141**	5292	11,047	4249	4744
ER465	48,627	44,255	35,882	**19,952**	45,485	58,669	46,125	37,479
ER940	234,479	229,576	140,222	**114,166**	147,677	234,354	208,462	136,475
ER2343	2,011,122	2,009,132	1,608,450	**1,606,656**	1,886,728	2,102,322	2,047,332	1,788,979
BA500	997	892	222	**203**	240	2601	269	238
BA1000	3770	3057	622	**580**	643	15,940	687	717
BA2500	31,171	28,044	4311	4292	4464	46,666	**4254**	4677
BA5000	170,998	146,753	12,273	12,273	12,769	322,990	**11,886**	13,527
WS250	14,251	**13,786**	16,110	16,110	16,110	16,110	16,110	16,110
WS500	54,201	**53,779**	55,491	55,163	67,162	69,379	68,638	70,125
WS1000	311,700	**308,596**	319,600	319,600	319,600	318,801	319,600	319,600
WS1500	717,369	703,241	716,509	**653,015**	759,528	749,716	757,066	759,528
FF250	1841	1386	313	**302**	458	676	351	333
FF500	2397	1904	390	**344**	537	1754	622	360
FF1000	92,800	59,594	2167	**1880**	2806	10,289	6472	2137
FF2000	387,248	256,905	7647	**7432**	8272	44,505	9983	9083
Scores	0	3	0	**11**	0	0	2	0

Table 4.11: The objective values of the small networks after removing k-critical nodes calculated by each of the tested approaches

Problem	SA	PBIL	DFSH	DFSH-post	Deg	Close	Between	Page
ER235	38	84	0.004	0.004	0.002	0.009	0.013	0.003
ER465	110	183	0.01	0.013	0.003	0.022	0.031	0.007
ER940	361	469	0.015	0.017	0.004	0.11	0.13	0.011
ER2343	1931	2171	0.047	0.049	0.013	0.57	0.84	0.028
BA500	66	126	0.006	0.007	0.003	0.018	0.067	0.005
BA1000	172	264	0.012	0.013	0.006	0.065	0.15	0.01
BA2500	840	1178	0.029	0.031	0.018	0.38	0.65	0.023
BA5000	3154	3515	0.061	0.064	0.03	1.57	2.78	0.048
WS250	70	135	0.007	0.007	0.004	0.011	0.018	0.006
WS500	173	263	0.01	0.01	0.007	0.037	0.079	0.009
WS1000	548	676	0.019	0.02	0.01	0.16	0.24	0.013
WS1500	1816	2064	0.027	0.029	0.017	0.23	0.38	0.021
FF250	37	88	0.008	0.009	0.002	0.01	0.013	0.006
FF500	156	233	0.013	0.014	0.003	0.021	0.077	0.009
FF1000	410	509	0.02	0.022	0.005	0.1	0.16	0.018
FF2000	1723	1961	0.034	0.035	0.008	0.39	0.57	0.032

Table 4.12: The runtime (in seconds) of the population based approaches, proposed heuristics, and centrality measures

The average runtime of the SA and PBIL approaches [72] and the runtime of the proposed heuristics and centrality measures are given in Table 4.12 for all network samples. As expected, the degree centrality has the lowest runtime among other non-population based approaches and the proposed heuristic is in the third place after the PageRank.

Discussion

DFSH-post showed a promising improvement on the quality of solution of *DFSH* in small benchmark networks presented in [72] as expected from the results of binomial test comparisons presented in previous section. Also, *DFSH-post* results in objective values that are less than the other tested approaches in most of the benchmark problems (11 out of 16 problems), and its runtime is the third fastest after degree centrality and PageRank approaches. According to the results given in this section, *DFSH-post* still performs well in unseen benchmark data which indicates that the weights of the proposed heuristic are properly tuned for tested network models.

Chapter 5

Real-world Networks

The proposed heuristic *DFSH-post* was evaluated in Chapter 4 using benchmark data generated by different network models. *DFSH-post* was also evaluated based on comparisons between the resulting objective values of *DFSH-post* and different centrality measures. Each of the tested network models attempted to model a specific behaviour observed in real-world networks, such as the power-law degree distribution or the small-world property. However, real-world networks may not have exactly the same characteristics as observed in the aforementioned benchmarks. For example, the WS model generates small-world networks with clustering coefficient around 0.45 while Watts and Strogatz presented a small-world Film actors network with 0.79 clustering coefficient [75]. In order to further evaluate the performance of *DFSH-post* and centrality measures on unseen data, 14 real-world networks of sizes varying from hundreds to millions of nodes are utilized. Table 5.1 represents the number of nodes, edges, average path length, clustering coefficient, average degree of nodes, initial cut vertices and initial bridges for each of the tested real-world networks. These networks represent real objects and the connections between them (e.g., communication networks, roads, diseases, and social networks). The employed real-world networks are described below with the applications of the CNDP in each of these networks.

The USAir97 network [61] represents the flight connections between major US airports in 1997. The nodes represent the airports, and there is a link between two nodes if a flight connection exists between those two cities. It is important to find critical airports in this network since the removal of those nodes minimizes the aerial access and consequently jam the aerial network, the aerial access is the pairwise connectivity between two cities as defined in the CNDP (Eq. (2.2)).

Three disease networks were proposed by Goh et al. [39] based on the relations between disease genes and genetic disorders. The Human-Disease network represents the relation between disorders, where the nodes are disorders and two nodes are connected if there is at least one gene that is implicated in both of them (e.g., breast cancer and prostate cancer are connected since three genes are implicated in both). The Gene-

Disease network contains the disease genes as nodes and two genes are connected if they are implicated in at least one common disorder. The Bipartite-Disease network shows the relation between genes and disorders, where the genes and disorders are represented as nodes and a gene v is connected to a disorder u if mutations on gene v cause the disorder u. Finding the critical nodes in these disease networks helps to point out which disorders or disease genes are most important to be cured.

The Hep-th (high energy physics theory) citation network [53] represents the citations between different papers, where nodes represent papers submitted from 1993 to 2003 and edges represent citations between them. Finding the critical nodes of the Hep-th citation network reveals the papers that are more influential in different subcommunities of the network.

The Email network [54] is a communication network, where each node is an email address and two nodes are connected if at least one email is exchanged between them. Eliminating the critical nodes in the Email network minimizes the damage of spreading viruses such as MyDoom [76] that spread via email.

The Internet network [11] represents the communications between about 125,000 computers, and there is a link between two computers if they communicate through the Internet. Finding critical nodes is important in order to prevent the spread of dangerous viruses through the Internet network.

The Marker, Youtube and LiveJournal networks [30, 31, 78] are on-line social networks that represent the friendship between users in an Israelis on-line social network website, the Youtube website and an on-line blogging community, respectively. In these social networks, the nodes represent the users and two nodes are connected if they add each other as friends. In order to prevent rumour spreading in these social networks, the most critical nodes (users) in the network should be determined and then prevented from spreading the rumour. Moreover, finding critical nodes in such social networks is important for the purpose of targeted marketing advertising.

The PA, EA, and CA Road networks [54] represent the road systems of Pennsylvania, Texas, and California, respectively, where the nodes represent the intersections and the roads connecting intersections to each other are represented as links. Blocking the most critical intersections of a state road network results in minimum pairwise connectivity between different areas of a state and consequently jams the traffic.

The Skitter network [53] is an Internet topology network gathered in 2005 that represents the connections between over 1.5 million routers in the Internet. The critical routers should be protected from viruses or worms in order to minimize their spread through the Internet. Moreover, in the sense of attacking the Internet, the viruses can attack the most critical routers in order to spread faster.

Network	n	m	D	C	$\langle k \rangle$	ζ	ξ
USAir97	232	1635	2.52	0.44	14.09	33	17
Human Disease	516	1188	6.508	0.43	4.604	111	112
Gene Disease	903	6760	5.933	0.84	14.972	57	107
Bipartite Disease	1723	1932	9.624	0.0003	2.242	1253	552
Hep-citation	24,402	332,014	3.314	0.048	27.212	1309	1203
Email	33,696	180,811	4.025	0.085	10.731	9682	1236
Marker	69,317	1,644,794	3.059	0.045	47.457	15,672	5629
Internet	124,651	193,620	11.277	0.038	3.106	55,455	37,656
Youtube	1,134,890	2,987,624	5.279	0.006	5.265	667,090	223,409
PA Road	1,087,562	1,541,514	307.975	0.059	2.834	216,775	193,743
TX Road	1,351,137	1,879,201	415.713	0.06	2.781	286,621	256,484
CA Road	1,957,027	2,760,388	311.547	0.06	2.821	372,704	326,965
Skitter	1,694,616	11,094,209	5.074	0.005	13.093	231,165	111,350
Live Journal	3,997,962	34,681,189	5.57	0.125	17.349	821,887	594,079

Table 5.1: The basic characteristics of 14 real-world networks. The measured quantities are: number of vertices n, number of edges m, average path length D, clustering coefficient C, average degree $\langle k \rangle$, number of bridges ζ, and number of cut vertices ξ.

As mentioned in Chapter 4, the purpose of tuning the weights of the proposed heuristic was to be able to run *DFSH-post* on unseen networks and gain similar performance as observed in the benchmark data without re-tuning. However, the properties of tested real-world networks are significantly different than what is observed in the benchmarks generated by various network models that were presented in Chapter 4. The weights tuned for BA, ER, and WS networks are not suitable to be used for the utilized real-world networks, the reasons are given in below.

As shown in Table 5.1, there exist cut vertices and bridges in all tested real-world networks, but no cut vertex or bridge exists in the networks generated by WS model, and BA-m2 networks do not contain any bridges as well. The BA-m1 networks have zero clustering coefficient ($C = 0$) while the clustering coefficient for all real-world networks is $C > 0$. Moreover, Newman [59] stated that the clustering coefficient of an ER network is

$$C = p = \frac{\langle k \rangle}{n} \tag{5.1}$$

where $\langle k \rangle$ is the average degree of the network and n is the size of the network, which tends to zero as $n \to \infty$ and $\langle k \rangle$ is a constant. Also, the ER networks do not have any of the common properties observed in real-world networks that were mentioned in Section 2.3, so the weights tuned for the ER networks are not likely to be suitable for any of the real-world networks.

Hence, the tuned weights for the FF model are the most suitable among other network models to be used for the real-world networks. Since the clustering coefficient and average degree of nodes for the FF networks are around $C \approx 0.2, \langle k \rangle \approx 3$ (shown in Table 4.6), the FF networks are also different than the real-world networks presented in Table 5.1.

The four small real-world USAir97, Human Disease, Gene Disease, and Bipartite Disease networks are plotted in Figures 5.1 through 5.4 in order to show the difference between the topology of the real-world networks and the network models presented in Section 4.1. As can be seen from Figures 5.1 through 5.4, the network models with the most similar topologies to the real-world networks are BA and FF models. Those network models produce tree-based networks with branches that start to grow as more nodes are added to the network (preferential attachment), while a number of connections between different branches exist in the networks shown in Figures 5.1 through 5.4.

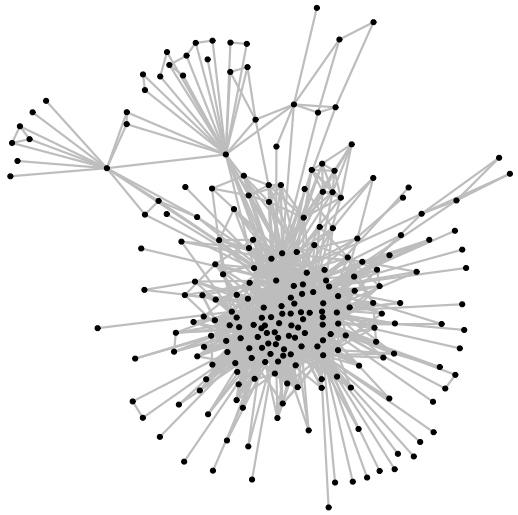


Figure 5.1: The USAir97 network with 232 vertices and 1635 edges

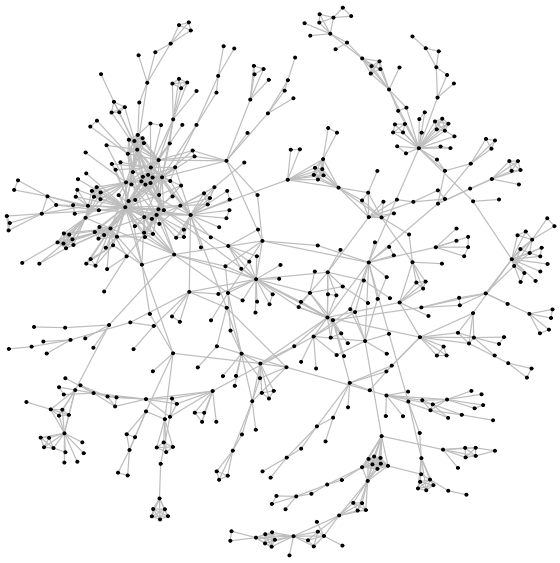


Figure 5.2: The Human Disease network with 516 vertices and 1188 edges

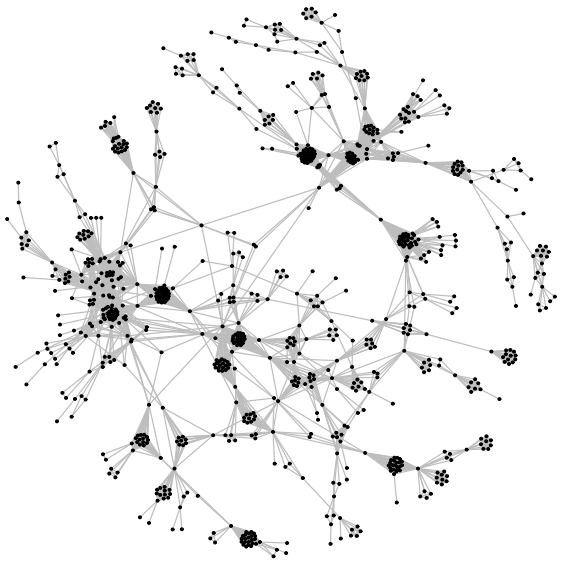


Figure 5.3: The Gene Disease network with 903 vertices and 6760 edges

Figure 5.4: The Bipartite Disease network with 1723 vertices and 1932 edges

The distribution of weights in different network models (given in Section 4.2) showed the effect of selecting a good set of weights on the performance of *DFSH-post*. As stated above, the most suitable weights between the network models presented in Section 4.1 to be used for the real-world networks are those of the FF model. However, in order to show the quality of *DFSH-post* when the sets of weights are tuned particularly for a network, the weight tuning procedure for *DFSH-post* was performed on the four small size real-world networks (the USAir97 and three disease networks), i.e., re-tuned and previously tuned results will be shown. Table 5.2 shows determined weights for the USAir97, Human Disease, Gene Disease, and Bipartite Disease networks. According to the calculated weights, it can be concluded that cut vertices and bridges are more important with regard to the objective of the CNDP since their calculated weights are higher than the other two weights for local bridges and regular nodes.

The effect on the objective value by the newly tuned weights given in Table 5.2 and the weights tuned for the FF model are shown in Table 5.3 for the four small real-world networks. As can be seen in Table 5.3, the re-tuned weights result in lower objective values than the weights tuned for the FF model.

	USAir97					Human Disease				
k-value	w_1	w_2	w_3	w_4	θ	w_1	w_2	w_3	w_4	θ
1%	0	0	0	0.4	0.5	0	0.25	0	0.4	0.4
10%	0.1	0.1	0.7	0.1	0.5	0.1	0.4	0.55	1	0.4
20%	0.1	0.1	0.55	0.85	0.9	0.1	0.4	1.45	0.7	0.55
30%	0.1	1	1.6	1.6	0.9	0.1	1.45	0.85	0.25	0.6
40%	0.1	1.6	0.1	0.4	0.85	0.1	0.7	0.1	1	0.65
50%	0.1	0.1	0.55	1.45	0.9	0.1	0.85	0.85	1	0.95
	Gene Disease					Bipartite Disease				
	w_1	w_2	w_3	w_4	θ	w_1	w_2	w_3	w_4	θ
1%	0	0.25	0.1	0.85	0.5	0.1	0.25	0.7	0.85	0.3
10%	0.1	1.6	1.45	1.15	0.6	0.25	0.4	0.25	0.55	0.3
20%	0.1	0.25	1	0.7	0.65	0.1	0.25	0.55	0.85	0.6
30%	0.25	0.4	0.4	0.55	0.6	0.1	0.1	0.85	0.4	0.65
40%	0.25	0.7	0.1	0.25	0.6	0.1	0.1	0.7	0.85	1
50%	0.1	0.1	1	1.6	0.6	0.1	0.25	0.85	0.4	1

Table 5.2: The set of weights and threshold that had the lowest objective value among other tested weights per each k-value for the USAir97 and three disease networks

The distribution of objective values of all tested sets of weights are plotted in order to show the effect of tuning the weights on the performance of the proposed heuristic. The distribution of objective values on weights when k = 1% is shown in Figure 5.5 for the four small real-world networks, and the plots of the distribution of objective values for other k-values are given in Appendix C.

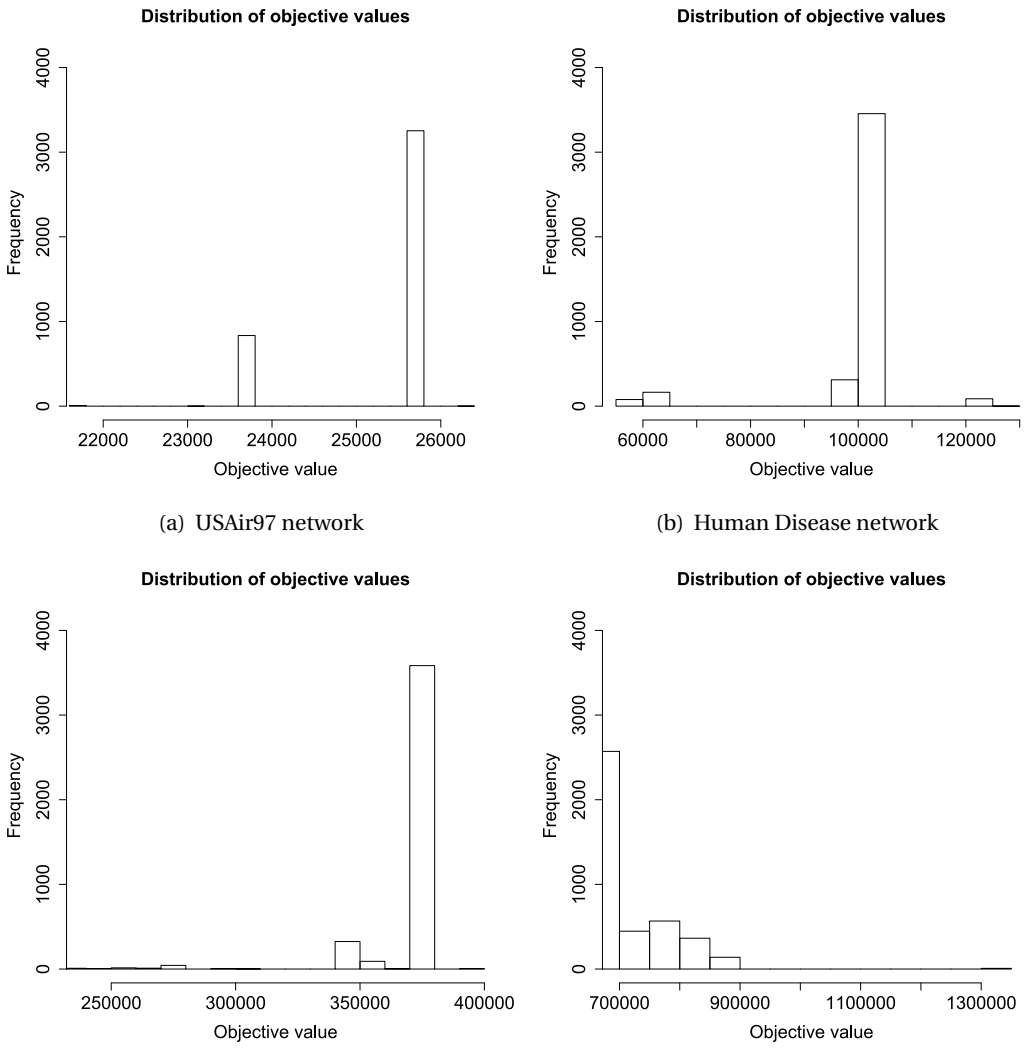
As can be seen in Figure 5.5, only a few sets of weights result in the minimum objective value among the other tested weights for the USAir97, Human Disease, and Gene Disease networks. Moreover, there exist sets of weights that result in objective values almost two times higher than the lowest value found in tested weights for the USAir97, Human Disease, and Gene Disease networks. Hence, selecting proper weights has a high effect on the quality of solution of the proposed heuristic for the USAir97, Human Disease, and Gene Disease networks. As can be seen in Figure 5.5(d), a few sets of weights result in objective values more than twice as high as the lowest objective value found from tested sets of weights in the Bipartite Disease network. Therefore, selecting proper weights of the proposed heuristic is considered an easier problem for Bipartite Disease network than the other three networks. The distributions of objective values for higher k-values (given in Appendix C) showed that the objective values calculated by tested weights are in a long range. For example, the lowest objective value found from tested weights for the USAir97 network is 7771 when k = 10%, while the highest objective value found from tested weights is 18,732 which is about three times higher than what the heuristic can actually achieve. Based on the comparisons between the objective val-

ues resulted by re-tuned weights for the four small real-world networks and the weights tuned for the FF model, it can be concluded that selecting proper sets of weights may improve the performance of the proposed heuristic high enough to outperform other approaches that were tested in this thesis. For example, by using the weights tuned for the FF model, the proposed heuristic loses to other approaches in 4 out of 6 different k-values, while the heuristic with re-tuned weights results in minimum objective value in all tested k-values.

The effect on the objective values after removing different k number of nodes by *DFSH-post* and centrality measures are calculated for each of the real-world networks. The weights tuned for the FF model are used in *DFSH-post* for the real-world networks. The procedures to calculate the outputs of centrality measures that take longer than four hours were excluded from comparisons due to limited available resources. The runtime (in seconds) of the approaches for the tested real-world networks is given in Table C.15. As can be seen in Table C.15, the closeness and betweenness centrality measures are excluded from the experiments for the real-world networks of size one million or larger since their runtime takes more than four hours (the run time of the closeness for the PA Road network is actually about 28 hours). The runtime for *DFSH-post* is less than an hour for networks with up to two million nodes and three million edges, and it is the third fast approach after degree centrality and PageRank. The effect on the objective values by the tested approaches for all of the tested real-world networks and k-values are given in Appendix C. The approaches that result in the lowest objective value among other tested approaches are reported in Table 5.4 per each k-value for the real-world networks. Since the closeness and betweenness were excluded from comparisons for the real-world networks of size larger than one million, the comparisons were only between *DFSH-post*, degree centrality and PageRank.

	USAir97		Human Disease		Gene Disease		Bipartite Disease	
k-value	*new*	FF	*new*	FF	*new*	FF	*new*	FF
1%	**21,749**	25,651	**55,362**	104,415	**247,067**	372,077	**697,455**	714,031
10%	**7771**	**7771**	**1327**	1551	**7544**	8603	**1532**	3013
20%	**415**	2425	**397**	415	**3331**	3395	**212**	247
30%	**119**	1109	**194**	202	**2260**	2363	**31**	85
40%	**42**	54	**90**	97	**1432**	1533	**1**	11
50%	**5**	472	**28**	69	**773**	4747	**0**	**0**

Table 5.3: The objective values of the USAir97 and three disease networks calculated by the re-tuned weights and the weights tuned for the FF model per each k-value. The lower objective value is bolded per k-value for each of the real-world networks.



(a) USAir97 network

(b) Human Disease network

(c) Gene Disease network

(d) Bipartite Disease network

Figure 5.5: The distribution of objective values across all weights are plotted for the four small real-world networks USAir97, Human Disease, Gene Disease, and Bipartite Disease networks when $k = 1\%$

k	USAir97	Human-D	Gene-D	Bipartite-D	Hep-citation	Email	Marker
1%	*Between*	*Between*	*Between*	Page, Deg	Between	Between	Between
10%	*DFSH*	*DFSH*	*DFSH*	*Deg*	Between	Deg	Between
20%	*Page*	*DFSH*	*DFSH*	*DFSH*	Between	Page	Page
30%	*Page*	*DFSH*	*Page*	*DFSH*	Between	*DFSH*	Page
40%	*DFSH*	*DFSH*	*Page*	*Page*	DFSH	*DFSH*	Between
50%	*Page*	*DFSH*	*Page*	all except Close	Page	*DFSH*	*DFSH*
	Internet	Youtube	PA Road	TX Road	CA Road	Skitter	L-Journal
1%	Page	Page	Page	Page	Page	Page	Page
10%	Deg	Page	Deg	Deg	Deg	Page	Page
20%	Page	Page	Deg	Deg	Deg	Page	Page
30%	*DFSH*	Page	Deg	Deg	Deg	*DFSH*	Page
40%	*DFSH*	Page	*DFSH*	*DFSH*	*DFSH*	*DFSH*	Deg
50%	*DFSH*	*DFSH*, Deg	*DFSH*	*DFSH*	*DFSH*	*DFSH*	*DFSH*

Table 5.4: The approach that results in the lowest objective value among other tested approaches per k-value in the tested real-world networks, *DFSH-post* heuristic is abbreviated as *DFSH* in this table.

DFSH-post	Closeness	Betweenness	Degree	PageRank
31	0	12	16	30

Table 5.5: The number of times each approach results in the lowest objective value per k-value for the tested real-world networks, the total number of cases is 84.

The number of times that each approach results in the lowest objective value among the other approaches for the tested real-world networks is shown in Table 5.5. Based on the results given in Table 5.5, the proposed methodology wins at 31 cases out of the total 84 cases, which is bolded as the winner with highest number of times that result in the lowest objective value (it won one more than PageRank). Similar to the results of the benchmarking experiments, the closeness centrality did not outperform other approaches in any of the tested cases. The betweenness centrality measure won at 12 cases of the real-world networks of size smaller than one million nodes (its value was not calculated for the larger networks). As shown in Table C.15, the degree centrality is the fastest approach among others, its run time for the Live Journal network was less than 3 seconds. However, the degree centrality results in the lowest objective value in only 16 cases, which puts it in the third place as shown in Table 5.5. The PageRank was the best among other centrality measures by resulting in the lowest objective value in 30 cases and being the second fastest approach after the degree centrality. Moreover, since the difference between *DFSH-post* and PageRank in the number of times that each of them results in minimum objective value is only one case, the PageRank is fairly comparable to *DFSH-post* in case of quality of solution for the real-world networks.

Discussion

The effect on the objective value after removing nodes selected by *DFSH-post* was compared to the other centrality measures in 14 real-world networks, and the results suggest that *DFSH-post* wins at highest number of cases among the other approaches (in 31 out of 84 cases), although the PageRank results in the lowest objective value at 30 cases that is close to *DFSH-post*.

The weights of *DFSH-post* are re-tuned for four small networks. The results of comparisons between the effect on the objective values by new weights and the weights of FF model show that the proposed heuristic results in minimum objective value in all cases when the weights are properly tuned. In conclusion, although *DFSH-post* has the highest number of times that its objective value is the lowest, it can still perform better when the weights are properly tuned.

Chapter 6

Conclusions and Future Work

The critical node detection problem is an NP-complete problem, which means that no polynomial algorithm exists to solve it, unless $P = NP$. Therefore, heuristics need to be developed to find good solutions. The CNDP is of particular interest because of the growing attention to networks in real-world and to solve this problem for them (e.g., finding the most critical members of a terrorist communication network). The CNDP has many real-world applications, and some of them (such as social networks) grow over time, their sizes may exceed millions of nodes. Previous works on the CNDP did not focus on very large network data. The aim of this thesis was to design fast and efficient heuristics applicable on large networks. A DFS-based heuristic was designed to crawl the network to gather information about the nodes in order to use them for ranking the nodes and then select the k highest ranked nodes as the k critical nodes. Since the proposed DFS-based heuristic was not able to outperform other approaches in most of the cases, a post-processing procedure was developed to boost its performance. The proposed post-processing procedure deselects nodes where the number of their neighbours is greater than a given threshold θ.

In order to assess the performance of the proposed heuristic with post-processing procedure *DFSH-post*, various benchmark suites containing networks of different sizes were generated by the network models presented in Section 4.1. Each of the network models has different characteristics and topologies, and the weights and threshold θ of the *DFSH-post* were tuned for each network model based on experiments on the generated benchmark suites. After tuning the weights, the performance of the *DFSH-post* was compared to the centrality based approaches presented in Section 2.5, and the results of comparisons showed that the *DFSH-post* is the best approach since it won in 23 out of total 30 cases. Moreover, the results of the proposed post-processing procedure were compared to the DFS-based heuristic based on binomial tests on the objective values calculated by them for all of the generated benchmark suites. The results of comparisons showed that *DFSH-post* has either the same performance as *DFSH* or better than it in all tested benchmark networks. The performance of the centrality based approaches

and proposed heuristic were also compared to the population based approaches SA and PBIL introduced by Ventresca [72] for the 16 benchmarks presented in [72], and *DFSH-post* was again the winner by resulting in the lowest objective value in 11 out of the total 16 cases.

Moreover, *DFSH-post* and proposed centrality based approaches were compared to each other in 14 real-world networks. Based on the characteristics of the real-world networks, it was concluded that the weights tuned for the FF model are the most suitable among other network models to be used for *DFSH-post*. In order to show the quality of *DFSH-post* when proper sets of weights are selected, the weights were re-tuned by experiments for four small real-world networks. The closeness and betweenness centrality measures were unable to finish their calculations in proper time (less than 4 hours) for networks with more than one million nodes, and therefore they were disregarded in comparisons for those networks. *DFSH-post* was again the winner approach among others by resulting in the lowest objective value in 31 out of the total 84 cases. In conclusion, the proposed methodology showed better performance than centrality measures in both of the benchmark data and real-world networks.

The critical node detection problem has future works in both theoretical and experimental aspects. In the sense of designing heuristics more powerful than the one presented in this thesis, designing a fast algorithm that is able to find the next selected node based on the effect of removing the previously selected nodes from the graph may improve the results. Similar works to this algorithm was done by Holme et al. [42] by recalculating the betweenness and degree centrality values of the nodes after removing a node from the network. Their results showed that the recalculated centrality measures had better performance than the regular versions in the tested networks [42]. An easy way to do this recalculation is to perform the heuristic after deleting each node and it takes approximately k-times longer to complete the ranking procedure.

The weights of the heuristic tuned for different network models were unable to result in minimum objective value among other approaches for real-world networks in many cases (in 53 out of 84 cases). The reason is that the characteristics of tested network models are different than what is observed in real-world networks, they are different in clustering coefficient, number of edges, etc. Therefore, it is suggested to find other network models that have topological features more similar to real-world networks and tune the weights of the proposed heuristic for them. Moreover, a faster heuristic that can compete with degree centrality and PageRank needs to be developed.

Although tuning the weights in the proposed heuristic helps to maintain the quality of solution of the algorithm for different network topologies, it needs time for computing proper weights. Therefore, future works also include developing heuristics that do not need weight tuning procedure and their quality of solution is competitive to the proposed heuristic.

Since the CNDP is only formally defined for unweighted and undirected networks, there is a lot of room to define the problem on weighted, directed, or even weighted directed networks, and then design and develop proper heuristics for them.

Bibliography

[1] R. Albert and A. L. Barabasi. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, June 2002.

[2] R. Albert, H. Jeong, and A. L. Barabasi. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.

[3] M. Arita. The metabolic world of escherichia coli is not small. *Proceedings of the National Academy of Sciences of the United States of America*, 101:1543–1547, February 2004.

[4] A. Arulselvan, C. W. Commander, L. Elefteriadou, and P. M. Pardalos. Detecting critical nodes in sparse graphs. *Computers and Operations Research*, 36:2193–2200, July 2009.

[5] A. Arulselvan, C. W. Commander, P M. Pardalos, and O. Shylo. Managing network risk via critical node identification. *Gulpinar, N., Rustem, B. (eds.) Risk Management in Telecommunication Networks*, 2010.

[6] A. Arulselvan, C. W. Commander, O. Shylo, and P. M. Pardalos. Cardinality-constrained critical node detection problem. In *Performance Models and Risk Management in Communications Systems*, volume 46, pages 79–91. Springer New York, 2011.

[7] B. Balasundaram, S. Butenko, and S. Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10:23–39, 2005.

[8] A. Barabasi, R. Albert, and H. Jeong. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1-4):69–77, June 2000.

[9] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[10] R. Barnes. *An Algorithm for Partitioning the Nodes of a Graph*. Research Report. IBM Thomas J. Watson Research Center. Center, 1981.

[11] V. Batagelj and A. Mrvar. The Internet network. Freely available at Pajek datasets. `http://vlado.fmf.uni-lj.si/pub/networks/data/web/web.zip`.

[12] M. Bellare, O. Goldreich, and M. Sudan. Free bits, pcps and non-approximability - towards tight results. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, FOCS '95, pages 422–431. IEEE Computer Society, 1995.

[13] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424:175–308, 2006.

[14] V. Boginski and C. W. Commander. Identifying critical nodes in protein-protein interaction networks. In S. I. Butenko, W. A. Chaovilitwongse, and P. M. Pardalos, editors, *Clustering Challenges in Biological Networks*, pages 153–167. World Scientific, 2008.

[15] B. Bollobas and O. Riordan. *Mathematical Results on Scale-Free Random Graphs*, pages 1–37. Wiley–WCH, 2002.

[16] B. Bollobás and O. Riordan. *Percolation*. Cambridge University Press, 2006.

[17] S. P. Borgatti. Identifying sets of key players in a social network. *Comput. Math. Organ. Theory*, 12:21–34, April 2006.

[18] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.

[19] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, April 1998.

[20] R. Cohen, K. Erez, D. ben Avraham, and Sh. Havlin. Resilience of the internet to random breakdowns. *Phys. Rev. Lett.*, 85:4626–4628, November 2000.

[21] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. MIT Press, 2009.

[22] P. Crescenzi, R. Silvestri, and L. Trevisan. To weight or not to weight: where is the question? In *Proceedings of the 4th IEEE Israel Symposium on Theory of Computing and Systems*, pages 68–77, 1996.

[23] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda. Efficiency of scale-free networks: Error and attack tolerance. *PHYSICA A*, 320:622–642, March 2003.

[24] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda. Error and attack tolerance of complex networks. *PHYSICA A*, 340:388–394, 2004.

[25] K. Dasgupta, R. Singh, B. Viswanathan, D. Chakraborty, S. Mukherjea, A. Nanavati, and A. Joshi. Social ties and their relevance to churn in mobile telecom networks. In *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*, pages 668–677. ACM, 2008.

[26] M. Di Summa, A. Grosso, and M. Locatelli. Complexity of the critical node problem over trees. *Computers and Operations Research*, 38(12):1766–1774, 2011.

[27] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Structure of growing networks with preferential linking. *Physical Review Letters*, 85(21):4633–4636, November 2000.

[28] P. Erdős and A. Rényi. On random graphs. I. *Publ. Math. Debrecen*, 6:290–297, 1959.

[29] R. Ferrer i Cancho, C. Janssen, and R. V. Solé. The topology of technology graphs: Small world patterns in electronic circuits. *Physical Review E*, 64(4):046119–1–046119–5, October 2001.

[30] M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach, and Y. Elovici. Link prediction in social networks using computationally efficient topological features. In *Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Confernece on Social Computing (SocialCom)*, pages 73–80. IEEE, 2011.

[31] M. Fire, L. Tenenboim, R. Puzis, O. Lesser, L. Rokach, and Y. Elovici. Computationally efficient link prediction in variety of social networks. *ACM Transactions on Intelligent Systems and Technology*, 2013.

[32] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[33] S. Fortunato and C. Castellano. Community structure in graphs, December 2007.

[34] M.L. Fredman and R.E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Foundations of Computer Science, IEEE Annual Symposium on*, 0:338–346, 1984.

[35] L. C. Freeman, D. Roeder, and R. R. Mulholland. Centrality in social networks: Ii. experimental results. *Social Networks*, 2:119–141, 1980.

[36] A. M. Frieze and M. Jerrum. Improved approximation algorithms for max k-cut and max bisection. In *Proceedings of the 4th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 1–13. Springer-Verlag, 1995.

[37] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63. ACM, 1974.

[38] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995.

[39] K. I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A. L. Barabasi. The human disease network. *Proceedings of the National Academy of Sciences of the United States of America*, 104(21):8685–8690, May 2007.

[40] V. Hatzimanikatis, C. Li, J. Ionita, C. S. Henry, M. D. Jankowski, and L. J. Broadbelt. Exploring the diversity of complex metabolic networks. *Bioinformatics*, 21:1603–1609, 2005.

[41] B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. In *Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM)*, number 28 in Supercomputing '95. ACM, 1995.

[42] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han. Attack vulnerability of complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 65, May 2002.

[43] P. Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[44] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, October 2000.

[45] H. Jhuge and J. Zhang. Topological centrality and its e-science applications. *Journal of the American Society for Information Science and Technology*, 61:1824–1841, 2010.

[46] M. Jorgic, I. Stojmenovic, M. Hauspie, and D. Simplot-Ryl. Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks. In *Proceedings of the 3rd IFIP Mediterranean Ad Hoc Networking Workshop*, pages 360–371, 2004.

[47] V. Kann, S. Khanna, J. Lagergren, and A. Panconesi. On the hardness of approximating max k-cut and its dual. Technical report, 1997.

[48] I. Kanovsky. Small world models for social network algorithms testing. *Procedia Computer Science*, 1(1):2341–2344, 2010.

[49] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.

[50] A. Karygiannis, E. Antonakakis, and A. Apostolopoulos. Detecting critical nodes for manet intrusion detection systems. In *Proceedings of the Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, pages 7–15. IEEE Computer Society, 2006.

[51] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(1):291–307, 1970.

[52] K. Klemm and V.M. Eguiluz. Growing scale-free networks with small-world behavior. *Phys Rev E Stat Nonlin Soft Matter Phys*, 65(5 Pt 2):057102, 2002.

[53] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. pages 177–187. ACM Press, 2005.

[54] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Technical report, October 2008.

[55] J. Liu, H. A. Abbass, W Zhong, and D. G. Green. Local-global interaction and the emergence of scale-free networks with community structures. *Artificial Life*, 17(4):263–279, 2011.

[56] S. Milgram. The small world problem. *Psychology Today*, 61:60–67, 1967.

[57] J. C. Miller and J. M. Hyman. Effective vaccination strategies for realistic social networks. *Physica A: Statistical Mechanics and its Applications*, 386(2):780–785, December 2007.

[58] E. Nardelli, G. Proietti, and P. Widmayer. Finding the most vital node of a shortest path. *Theoretical Computer Science*, 296:167–177, March 2003.

[59] M. E. J. Newman. The structure and function of complex networks. *SIAM REVIEW*, 45:167–256, 2003.

[60] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), August 2003.

[61] Computational Analysis of Social and Organizational Systems. The USAir97 network. Freely available at. `http://www.casos.cs.cmu.edu/computational_tools/datasets/external/USAir97/index11.php`.

[62] J.P. Onnela, J. Saramaki, J. Hyvonen, G. Szabo, D. Lazer, K. Kaski, J. Kertesz, and A. L. Barabasi. Structure and tie strengths in mobile communication networks. *Proc. Natl. Acad. Sci. USA*, 104(18):7332–7336, 2007.

[63] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, 1998.

[64] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991.

[65] A. Pothen. Graph partitioning algorithms with applications to scientific computing. In *Parallel Numerical Algorithms*, pages 323–368. Kluwer Academic Press, 1997.

[66] S. Sahni and T. Gonzalez. P-complete approximation problems. *J. ACM*, 23(3):555–565, July 1976.

[67] L. A. Schintler, S. P. Gorman, A. Reggiani, R. Patuelli, A. Gillespie, P. Nijkamp, and J. Rutherford. Complex network phenomena in telecommunication systems. *Networks and Spatial Economics*, 5:351–370, 2005.

[68] C. Scoglio, W. Schumm, P. Schumm, T. Easton, S. Roy Chowdhury, A. Sydney, and M. Youssef. Efficient Mitigation Strategies for Epidemics in Rural Regions. *PLoS ONE*, 5(7):e11569, July 2010.

[69] M. Sheng, J. Li, and Y. Shi. Critical nodes detection in mobile ad hoc network. *Advanced Information Networking and Applications, International Conference on*, 2:336–340, 2006.

[70] T. Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.*, 5:1–34, 1948.

[71] L. Trevisan, G. B. Sorkin, M. Sudan, and D. P. Williamson. Gadgets, approximation, and linear programming. *SIAM J. Comput.*, 29(6):2074–2097, April 2000.

[72] M. Ventresca. Global search algorithms using a combinatorial unraking-based problem representation for the critical node detection problem. *Computers and Operations Research*, 2012.

[73] A. Wagner and D. A. Fell. The small world inside large metabolic networks. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1478):1803–1810, 2001.

[74] X. F. Wang and G. Chen. Complex networks: small-world, scale-free and beyond. *Circuits and Systems Magazine, IEEE*, 3(1):6–20, 2003.

[75] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.

[76] F-secure website. Email-Worm:W32/Mydoom.B. `http://www.f-secure.com/v-descs/mydoom_b.shtml`.

[77] K. Wehmuth and A. Ziviani. Distributed location of the critical nodes to network robustness based on spectral analysis. In *LANOMS*, pages 1–8, October 2011.

[78] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *ICDM*, pages 745–754, 2012.

[79] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[80] K. Zhao, A. Kumar, T. P. Harrison, and J. Yen. Analyzing the resilience of complex supply network topologies against random and targeted disruptions. *IEEE Systems Journal*, 5(1):28–39, 2011.

[81] T. Zhou, L. Lu, and Y. Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71:623, 2009.

Appendix A

Earlier designed heuristics and results of experimental comparisons

A.1 Earlier designed approaches

A.1.1 RANKH-prev1

The aim of this ranking function is to rank the nodes that have a vertex attribute (cut vertex) higher than other nodes without that vertex attribute. The ranking function was designed so that the ranks assigned to cut vertices are higher than the nodes that belong to a local bridge, then regular nodes, and finally the rank of nodes of degree one which is zero. In other words, a cut vertex will have higher rank than any node that is not a cut vertex, and an endpoint of a local bridge receives higher rank than other nodes that are not cut vertices. In order to rank between cut vertices, the value $\lambda_{i,G}$ was used, which is defined in Section 3.2. The ratio of the node's degree to the maximum degree of nodes was used to rank between endpoints of local bridges. For a regular node that has degree higher than 1, the maximum vertex similarity value between that node and its neighbours was used to rank between nodes that are neither a cut vertex nor an endpoint of a local bridge. The *RANKH-prev1* ranking function is formulated as:

$$RANKH\text{-}prev1(i) = \begin{cases} 2 + (1 - \lambda_{i,G}) & \text{if } i \text{ is a cut vertex} \\ 1 + \dfrac{deg(i)}{\Delta(G) + 1} & \text{if } i \text{ is not a cut vertex and is an endpoint of a local bridge} \\ 0 & \text{if } i \text{ has degree 1} \\ \underset{j \in \Gamma(i)}{max}\Big(1 - V_S(i,j)\Big) & \text{otherwise} \end{cases} \tag{A.1}$$

where $\lambda_{i,G}$ and $\Delta(G)$ have the same definition as given in Section 3.2.

As mentioned earlier, the value $(1-\lambda_{i,G})$ is in the range $(0,1]$ and by adding 2 to this value the range will be changed to $(2,3]$. The value $\frac{deg(i)}{\Delta(G)+1}$ is also in the range $(0,1]$ and its range is changed to $(1,2]$ after adding 1 to it. The value $(1-V_S(i,j))$ is in the range $(0,1]$ since the vertex similarity value between two nodes is in the range $(0,1]$ for non-local bridges. Therefore, all nodes of a graph G are ranked in the range $(0,3]$ except for nodes of degree 1 that are assigned to rank 0.

The main problem of this ranking function is that not always the k most critical nodes are all cut vertices, or even endpoints of local bridges. As an example, two nodes are marked with minus and plus signs in Figure 3.1 (in the left side of the graph) that are cut vertices and both have degree 2. After removing the node with minus sign from the graph, a disconnected component containing the node with plus sign and its neighbour is produced, where both of them have degree 1, and therefore it is not logical to always select all cut vertices first since there might exist other nodes in the graph that their removal decreases the objective value of the induced subgraph much more than the nodes similar to the one with plus sign in Figure 3.1.

A.1.2 RANKH-prev2

This ranking function was designed in order to assign higher ranks to cut vertices than other nodes, so an extra value $(\alpha+1)$ is added to the rank of each cut vertex to reach this goal, where α is the maximum rank of non-cut vertices. The purpose of this ranking function was to rank the nodes in a more complex way than *RANKH-prev1* and assign a score for each edge of a non-cut vertex. The sum notation in Eq. (A.2) adds all scores of the edges connecting any node i to its neighbours. The *RANKH-prev2* ranking function is formulated as:

$$\textit{RANKH-prev2}_i = \begin{cases} (\alpha+1)+(1-\lambda_{i,G}) & \text{if } i \text{ is a cut vertex} \\ -\infty & \text{if } i \text{ has degree 1} \\ \sum\limits_{j\in\Gamma(i)} \Big(\beta(i,j)\Big(w_2\frac{deg(i)}{\Delta(G)+1}\Big)+\big(1-\beta(i,j)\big)\Big(w_1\big(1-V_S(i,j)\big)\Big)\Big) & \text{otherwise} \end{cases} \tag{A.2}$$

where $\beta(i,j)$ is equal to 1 if there exists a local bridge between nodes i and j, and it is equal to 0 otherwise. In order to calculate the best set of undetermined weights w_1 and w_2, statistical experiments were done on different range of weights with the same experimental setup as given in Subsection 3.2.1, the results of these experiments are given in Section A.2.

This ranking function has the same deficiency as mentioned for the *RANKH-prev1*, since this ranking function assigns ranks to all cut vertices higher than other nodes in the graph, and there are situations that this strategy is not good enough (similar to the example given in Section A.1.1). The cut vertices are very important due to the fact that

their deletion disconnects the network, which reduces the objective value of the input graph, but in tree-based networks many cut vertices exist and an optimal answer may only contain a few cut vertices and the rest are endpoints of local bridges or regular nodes. Figure A.1 shows the nodes selected by an optimal solution and the *RANKH-prev2* in a Forest Fire network with 75 nodes and $k = 20$, the nodes are coloured in the same way as described for Figure 3.1. Two double circled optimal nodes in dark grey color that were only selected by the optimal solution are shown in Figure A.1 and none of them is a cut vertex, while the *RANKH-prev2* only selected cut vertices.

A.1.3 Previous post-processing procedures

Two post-processing procedures were designed prior to the one presented in Section 3.3, which aimed to penalize the neighbours of a vertex with the highest rank. The idea was to remove the node with the highest rank and give penalties to its neighbours since removing the highest ranked node has effect on its neighbours, at least their degree is reduced by 1. This post-processing procedure is shown in Algorithm 4.

Algorithm 4 Previous post-processing procedure

Require: Graph $G = (V, E)$
1: $L := \emptyset$
2: **while** $|L| < k$ **do**
3: pick node u with the highest rank in graph G
4: **for** $w \in \Gamma(u)$ **do**
5: Rank(w) =Rank(w)$-$*penalty*
6: **end for**
7: Rank(u) $= -\infty$
8: $L := L \cup \{u\}$
9: **end while**

The penalties used in *post-prev1* and *post-prev2* procedures are different, and they were compared based on experimental comparisons in order to determine which penalty strategy is better to be used. The *penalty* value for neighbours of a node v in *post-prev1* procedure is calculated by:

$$penalty = 1 - \frac{Rank(v)}{\lfloor max.rank \rfloor + 1} \tag{A.3}$$

and the penalty for a node v in *post-prev2* is calculated by:

$$penalty = \frac{Rank(v)}{\lfloor max.rank \rfloor + 1} \tag{A.4}$$

where *max.rank* is the highest rank found among all nodes in the given input graph G. The purpose of equations (A.3) and (A.4) is to penalize the neighbours of a node v based on its rank. In other words, the neighbours of a node that received the highest rank from

a ranking function will get the lowest penalty in *post-prev1* and the highest penalty in *post-prev2* procedure.

These post-processing procedures were unable to improve the results of ranking functions since the penalty value is the same for any k-value. A node which has 2 or 3 selected neighbours may receive too much penalties, while that node is still needed to be selected for a large k-value. A visual example is shown in Figure A.1, where the black node in double circles is connected to two other black nodes on its left side. The rank of the double circled black node is 2.3448 since $\lfloor \alpha \rfloor = 1$ and the $(1-\lambda_{i,G})$ value for that node is 0.3448. The two black coloured nodes connected to the double circled node from the left are also cut vertices, the minimum penalty of Eq. (A.4) for a cut vertex is 0.5 since minimum possible *Rank(v)* for a cut vertex in this graph is 2 and the highest possible value of *max.rank* is 3. After that *post-prev2* assigns minimum penalty 0.5 to the double circled node, its rank is reduced to 1.3448, which is less than all other cut vertices. The same problem occurs when using the penalty of *post-prev1*, since its minimum value is 0.5 for a cut vertex as well. Notice that in the induced subgraph after deletion of those two black coloured nodes that were neighbours of the double circled black node, the component containing that node will be separated into 3 components after its removal, which is considered a good choice in order to minimize the pairwise connectivity of the induced subgraph. It can be concluded that the *post-prev1* and *post-prev2* procedures are not suitable post processing procedures due to the fact that they penalize nodes more than expected. The results of experimental comparisons between previous ranking functions and post-processing procedures are given in the next section.

A.2 Detailed comparisons between earlier designed heuristics

In order to compare the negative approaches, the undetermined weights of *RANKH-prev2* are needed to be calculated per each dataset of benchmark networks. Table A.1 shows the best set of weights found for each k-value for all network models.

	BA-m1		BA-m2		WS		ER		FF	
k-value	w1	w2	w1	w2	w1	w2	w1	w2	w1	w2
1%	0	0.4	0	0.1	0.1	0	0.1	0	0	0.4
10%	0	0.4	0	0.25	0.2	0	0.25	0.7	0	0.4
20%	0	0.4	0	0.25	0.7	0.25	0	0.1	0	0.7
30%	0	0.4	0	0.25	0.85	0.55	0.1	0.85	0	0.1
40%	0	0.4	0	0.85	0.25	0.85	0.1	0.4	0	0.1
50%	0	0.4	0.1	0.85	0.25	0.4	0.25	0.55	0.85	0

Table A.1: The best set of weights of *RANKH-prev2* per each k-value for all benchmark models

Tables A.2 through A.6 show the results of comparisons between *RANKH-prev2* and *post-prev1* procedure for all network models. Tables A.7 through A.11 represent the comparisons between *RANKH-prev2* and *post-prev2* procedure for all network models. According to the results, neither *post-prev1* nor *post-prev2* procedure was able to outperform the *RANKH-prev2* on all cases. The results of comparisons between the *RANKH-prev1*, *RANKH-prev2*, and *DFSH-post* are presented in tables A.12 through A.16 for all network models. Based on the results *DFSH-post* wins in 28 cases out of 30.

k-value	wins	ties	losses	p-value
1%	68	30	2	~ 0
10%	99	1	0	~ 0
20%	100	0	0	~ 0
30%	98	2	0	~ 0
40%	0	100	0	1
50%	0	100	0	1

Table A.2: Number of wins, ties, and losses of the *RANKH-prev2* against *post-prev1* with the p-value of binomial tests in 2000 BA-m1 networks

k-value	wins	ties	losses	p-value
1%	94	6	0	~ 0
10%	100	0	0	~ 0
20%	100	0	0	~ 0
30%	100	0	0	~ 0
40%	99	1	0	~ 0
50%	100	0	0	~ 0

Table A.3: Number of wins, ties, and losses of the *RANKH-prev2* against *post-prev1* with the p-value of binomial tests in 2000 BA-m2 networks

k-value	wins	ties	losses	p-value
1%	91	9	0	~ 0
10%	100	0	0	~ 0
20%	100	0	0	~ 0
30%	100	0	0	~ 0
40%	99	1	0	~ 0
50%	98	2	0	~ 0

Table A.4: Number of wins, ties, and losses of the *RANKH-prev2* against *post-prev1* with the p-value of binomial tests in 2000 FF networks

k-value	wins	ties	losses	p-value
1%	0	100	0	1
10%	20	80	0	0.15
20%	92	8	0	~ 0
30%	98	2	0	~ 0
40%	99	1	0	~ 0
50%	97	3	0	~ 0

Table A.5: Number of wins, ties, and losses of the *RANKH-prev2* against *post-prev1* with the p-value of binomial tests in 2000 WS networks

k-value	wins	ties	losses	p-value
1%	0	21	0	1
10%	0	20	1	1
20%	0	20	1	1
30%	0	19	2	0.87
40%	0	19	2	0.87
50%	0	20	1	1

Table A.6: Number of wins, ties, and losses of the *RANKH-prev2* against *post-prev1* with the p-value of binomial tests in 420 ER networks

k-value	wins	ties	losses	p-value
1%	98	2	0	~ 0
10%	100	0	0	~ 0
20%	100	0	0	~ 0
30%	100	0	0	~ 0
40%	1	99	0	1
50%	1	99	0	1

Table A.7: Number of wins, ties, and losses of the *RANKH-prev2* against *post-prev2* with the p-value of binomial tests in 2000 BA-m1 networks

k-value	wins	ties	losses	p-value
1%	98	2	0	~ 0
10%	100	0	0	~ 0
20%	100	0	0	~ 0
30%	100	0	0	~ 0
40%	100	0	0	~ 0
50%	99	1	0	~ 0

Table A.8: Number of wins, ties, and losses of the *RANKH-prev2* against *post-prev2* with the p-value of binomial tests in 2000 BA-m2 networks

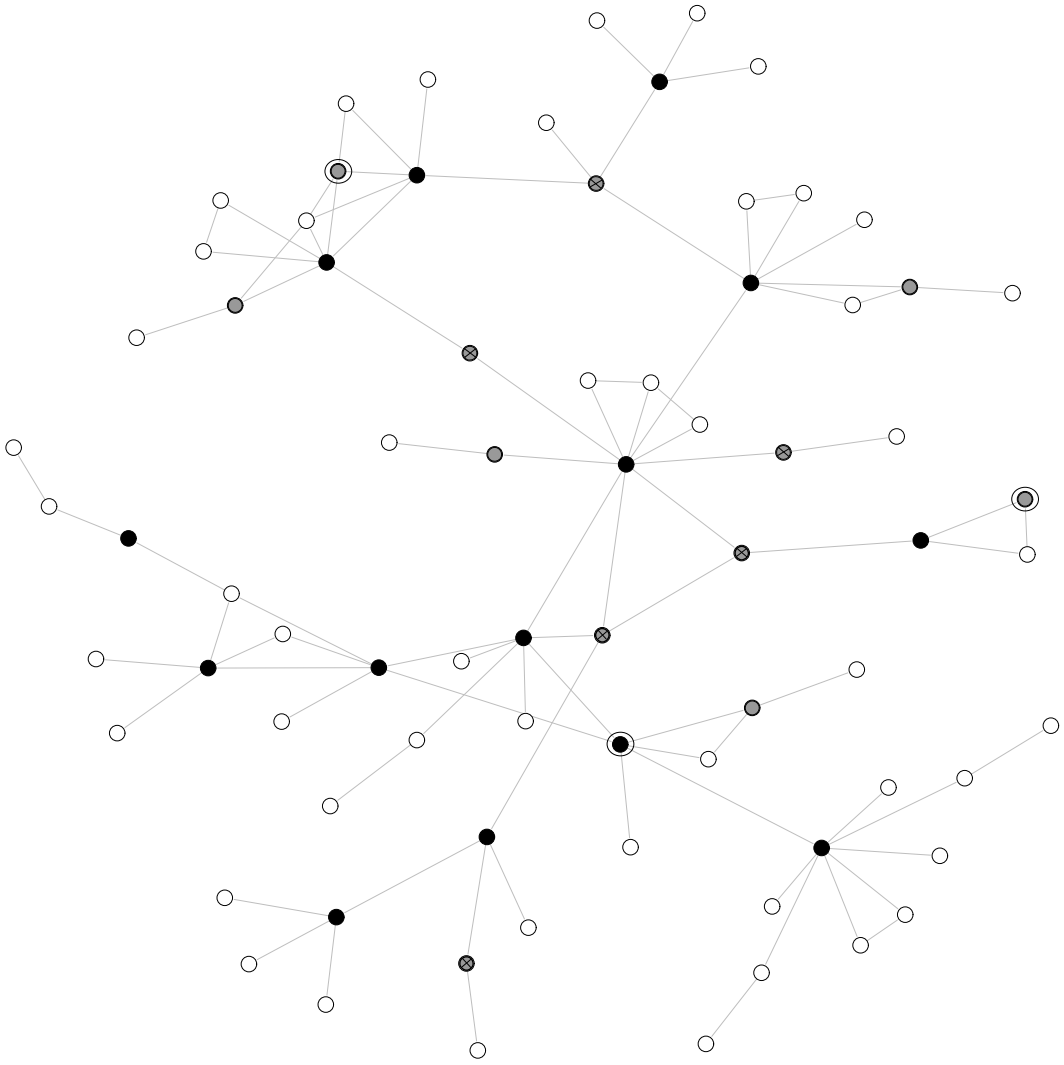


Figure A.1: A network sample with 75 nodes, where the shaded nodes represent the optimal and *RANKH-prev2* solutions. Black coloured nodes are in both solutions, while the grey and grey nodes with multiplication notation (×) only appeared in the optimal solution and *RANKH-prev2*, respectively.

k-value	wins	ties	losses	p-value
1%	94	6	0	~ 0
10%	100	0	0	~ 0
20%	100	0	0	~ 0
30%	100	0	0	~ 0
40%	100	0	0	~ 0
50%	0	1	99	~ 0

Table A.9: Number of wins, ties, and losses of the *RANKH-prev2* against *post-prev2* with the p-value of binomial tests in 2000 FF networks

k-value	wins	ties	losses	p-value
1%	0	100	0	1
10%	20	80	0	0.15
20%	92	8	0	~ 0
30%	98	2	0	~ 0
40%	0	8	92	~ 0
50%	32	68	0	0.016

Table A.10: Number of wins, ties, and losses of the *RANKH-prev2* against *post-prev2* with the p-value of binomial tests in 2000 WS networks

k-value	wins	ties	losses	p-value
1%	0	21	0	1
10%	0	20	1	1
20%	0	20	1	1
30%	0	19	2	0.87
40%	0	18	3	0.74
50%	0	19	2	0.87

Table A.11: Number of wins, ties, and losses of the *RANKH-prev2* against *post-prev2* with the p-value of binomial tests in 420 ER networks

k-value	*RANKH-prev1*	*RANKH-prev2*	*DFSH-post*
1%	100	100	100
10%	0	0	100
20%	0	0	100
30%	0	0	100
40%	100	100	100
50%	100	100	100

Table A.12: Number of wins of the *RANKH-prev1*, *RANKH-prev2* and *DFSH-post* on 100 BA-m1 network sizes, where each network size contains 20 network samples

k-value	*RANKH-prev1*	*RANKH-prev2*	*DFSH-post*
1%	45	2	69
10%	8	1	93
20%	0	0	100
30%	0	0	100
40%	0	0	100
50%	0	7	99

Table A.13: Number of wins of the *RANKH-prev1*, *RANKH-prev2* and *DFSH-post* on 100 BA-m2 network sizes, where each network size contains 20 network samples

k-value	*RANKH-prev1*	*RANKH-prev2*	*DFSH-post*
1%	1	1	99
10%	0	0	100
20%	0	0	100
30%	0	0	100
40%	0	0	100
50%	0	0	100

Table A.14: Number of wins of the *RANKH-prev1*, *RANKH-prev2* and *DFSH-post* on 100 FF network sizes where each network size contains 20 network samples

k-value	*RANKH-prev1*	*RANKH-prev2*	*DFSH-post*
1%	100	100	100
10%	6	70	99
20%	1	1	100
30%	5	0	99
40%	0	1	99
50%	0	0	100

Table A.15: Number of wins of the *RANKH-prev1*, *DFSH-post* and ALG-post on 100 WS network sizes where each network size contains 20 network samples

k-value	*RANKH-prev1*	*RANKH-prev2*	*DFSH-post*
1%	20	21	18
10%	17	20	18
20%	17	15	18
30%	13	15	17
40%	8	12	20
50%	3	5	20

Table A.16: Number of wins of the *RANKH-prev1*, *RANKH-prev2* and *DFSH-post* on 100 ER network sizes where each network size contains 20 network samples

Appendix B

Detailed results of benchmarks comparisons

The binomial test performs an exact test of the statistical significance of the difference between two categories A and B. The test requires as input the number of wins of each category, and the output is a p-value that indicates whether the difference between categories A and B is significant at the 5% level. When the calculated p-value is less than 0.05, it results that the category with higher number of wins is significantly (at the 5% level) better than another.

Tables B.1 through B.5 show the results of binomial tests between *DFSH* and *DFSH-post* for BA-m1, BA-m2, ER, WS, and FF network models, respectively.

k-value	wins	ties	losses	p-value
1%	0	100	0	1
10%	0	2	98	~ 0
20%	0	0	100	~ 0
30%	0	3	97	~ 0
40%	0	100	0	1
50%	0	100	0	1

Table B.1: Number of wins, ties, and losses of *DFSH* against *DFSH-post* with the p-value of binomial tests in 2000 BA-m1 networks

k-value	wins	ties	losses	p-value
1%	0	100	0	1
10%	0	19	81	~ 0
20%	0	0	100	~ 0
30%	0	0	100	~ 0
40%	0	0	100	~ 0
50%	0	29	71	~ 0

Table B.2: Number of wins, ties, and losses of *DFSH* against *DFSH-post* with the p-value of binomial tests in 2000 BA-m2 networks

k-value	wins	ties	losses	p-value
1%	0	21	0	1
10%	0	21	0	1
20%	0	20	1	1
30%	0	19	2	0.87
40%	0	18	3	0.74
50%	0	13	8	0.22

Table B.3: Number of wins, ties, and losses of *DFSH* against *DFSH-post* with the p-value of binomial tests in 420 ER networks

k-value	wins	ties	losses	p-value
1%	0	100	0	1
10%	0	100	0	1
20%	0	46	56	$2.07\ E^{-6}$
30%	0	5	95	$5.006\ E^{-24}$
40%	0	7	93	$3.45\ E^{-22}$
50%	0	1	99	$8.046\ E^{-29}$

Table B.4: Number of wins, ties, and losses of *DFSH* against *DFSH-post* with the p-value of binomial tests in 2000 WS networks

k-value	wins	ties	losses	p-value
1%	0	100	0	1
10%	0	3	97	~ 0
20%	0	0	100	~ 0
30%	0	0	100	~ 0
40%	0	0	100	~ 0
50%	0	0	100	~ 0

Table B.5: Number of wins, ties, and losses of *DFSH* against *DFSH-post* with the p-value of binomial tests in 2000 FF networks

Table B.6 shows the results of binomial tests between closeness and betweenness approaches for FF networks. Table B.7 shows the results between degree centrality and betweenness, the degree centrality is the winner for $k = 1\%, 10\%, 20\%, 40\%$ and the betweenness is the winner for $k = 30\%, 50\%$. Table B.8 shows the results of comparisons between winners of previous table and PageRank. Based on the results, degree centrality is the winner for $k = 1\%$, PageRank is the winner for $k = 10\%, 20\%, 30\%, 40\%$, and betweenness is the winner for $k = 50\%$. Table B.9 shows the results of comparisons between *DFSH-post* and the winner centrality measures from previous tables. *DFSH-post* wins in $k = 1\%, 10\%, 20\%, 30\%, 50\%$, the degree centrality wins in $k = 1\%$, and PageRank wins in $k = 10\%, 40\%$. When two approaches tie in the final binomial test, they both are considered as winner for that k-value.

k-value	wins	ties	losses	p-value
1%	0	4	96	~ 0
10%	0	0	100	~ 0
20%	0	0	100	~ 0
30%	0	0	100	~ 0
40%	0	0	100	~ 0
50%	0	0	100	~ 0

Table B.6: Number of wins, ties, and losses of closeness against betweenness with the p-value of binomial tests in 2000 FF networks

k-value	wins	ties	losses	p-value
1%	0	13	87	~ 0
10%	0	1	99	~ 0
20%	0	1	99	~ 0
30%	36	64	0	0.006
40%	1	87	13	0.419
50%	100	0	0	~ 0

Table B.7: Number of wins, ties, and losses of betweenness against degree centrality with the p-value of binomial tests in 2000 FF networks

k-value	wins	ties	losses	p-value
1%	93	7	0	~ 0
10%	0	4	96	~ 0
20%	0	0	100	~ 0
30%	0	0	100	~ 0
40%	0	0	100	~ 0
50%	75	25	0	~ 0

Table B.8: Number of wins, ties, and losses of betweenness and degree centrality against PageRank with the p-value of binomial tests in 2000 FF networks

k-value	wins	ties	losses	p-value
1%	1	90	9	0.61
10%	34	38	28	0.67
20%	44	56	0	0.001
30%	99	1	0	~ 0
40%	11	14	75	~ 0
50%	100	0	0	~ 0

Table B.9: Number of wins, ties, and losses of *DFSH-post* against other centrality measures with the p-value of binomial tests in 2000 FF networks

Table B.10 shows the results of binomial tests between closeness and betweenness approaches for BA-m1 networks, the betweenness is the winner for all k-values. Table B.11 shows the results between degree centrality and betweenness, the degree centrality is the winner for $k = 10\%$ and the betweenness is the winner for $k = 1\%, 20\%, 30\%$. Both of the approaches tie for $k > 40$ since their objective values are equal to zero in all cases. Table B.12 shows the results of comparisons between winners of previous table and PageRank. Based on the results, betweenness is the winner for $k = 1\%$ and PageRank is the winner for $k = 10\%, 20\%, 30\%$. The approaches tie for $k > 30\%$. Table B.13 shows the results of comparisons between *DFSH-post* and the winner centrality measures from previous tables. The betweenness wins in $k = 1\%, 40\%, 50\%$, *DFSH-post* wins in $k = 10\%, 20\%, 40\%, 50\%$, and PageRank wins in $k = 30\%, 40\%, 50\%$.

k-value	wins	ties	losses	p-value
1%	0	3	97	~ 0
10%	0	0	100	~ 0
20%	0	0	100	~ 0
30%	0	0	100	~ 0
40%	0	0	100	~ 0
50%	0	0	100	~ 0

Table B.10: Number of wins, ties, and losses of closeness against betweenness with the p-value of binomial tests in 2000 BA-m1 networks

k-value	wins	ties	losses	p-value
1%	89	11	0	~ 0
10%	0	3	97	~ 0
20%	97	3	0	~ 0
30%	97	3	0	~ 0
40%	0	100	0	1
50%	0	100	0	1

Table B.11: Number of wins, ties, and losses of betweenness against degree centrality with the p-value of binomial tests in 2000 BA-m1 networks

k-value	wins	ties	losses	p-value
1%	93	7	0	~ 0
10%	0	17	83	~ 0
20%	0	0	100	~ 0
30%	0	0	100	~ 0
40%	0	100	0	1
50%	0	100	0	1

Table B.12: Number of wins, ties, and losses of betweenness and degree centrality against PageRank with the p-value of binomial tests in 2000 BA-m1 networks

k-value	wins	ties	losses	p-value
1%	0	100	0	1
10%	99	1	0	~ 0
20%	51	49	0	~ 0
30%	0	1	99	~ 0
40%	0	100	0	1
50%	0	100	0	1

Table B.13: Number of wins, ties, and losses of *DFSH-post* against other centrality measures with the p-value of binomial tests in 2000 BA-m1 networks

Table B.14 shows the results of binomial tests between closeness and betweenness approaches for BA-m2 networks, the betweenness is the winner for all k-values. Table B.15 shows the results between degree centrality and betweenness, the degree centrality is the winner for all k-values. Table B.16 shows the results of comparisons between degree centrality and PageRank. Based on the results, degree centrality ties with PageRank for $k = 1\%$ and PageRank is the winner for $k > 1\%$. Table B.17 shows the results of comparisons between *DFSH-post* and PageRank. The degree centrality wins in $k = 1\%$, *DFSH-post* wins in $k = 1\%, 10\%, 20\%, 40\%$, and PageRank wins in $k = 1\%, 10\%, 30\%, 40\%, 50\%$.

k-value	wins	ties	losses	p-value
1%	0	4	96	~ 0
10%	0	0	100	~ 0
20%	0	0	100	~ 0
30%	0	0	100	~ 0
40%	0	0	100	~ 0
50%	0	0	100	~ 0

Table B.14: Number of wins, ties, and losses of closeness against betweenness with the p-value of binomial tests in 2000 BA-m2 networks

k-value	wins	ties	losses	p-value
1%	0	5	95	~ 0
10%	0	2	98	~ 0
20%	0	0	100	~ 0
30%	0	0	100	~ 0
40%	0	0	100	~ 0
50%	0	0	100	~ 0

Table B.15: Number of wins, ties, and losses of betweenness against degree centrality with the p-value of binomial tests in 2000 BA-m2 networks

k-value	wins	ties	losses	p-value
1%	0	94	6	0.71
10%	0	44	56	~ 0
20%	0	1	99	~ 0
30%	0	0	100	~ 0
40%	0	0	100	~ 0
50%	0	0	100	~ 0

Table B.16: Number of wins, ties, and losses of degree centrality against PageRank with the p-value of binomial tests in 2000 BA-m2 networks

k-value	wins	ties	losses	p-value
1%	0	90	10	0.513
10%	10	89	1	0.56
20%	33	67	0	0.013
30%	0	17	83	~ 0
40%	0	85	15	0.3
50%	0	0	100	~ 0

Table B.17: Number of wins, ties, and losses of *DFSH-post* against PageRank with the p-value of binomial tests in 2000 BA-m2 networks

Table B.18 shows the results of binomial tests between closeness and betweenness approaches for ER networks, they tie in all cases and betweenness is selected for next comparisons since it won in some network sizes. Table B.19 shows the results between degree centrality and betweenness, the approaches tie for all k-values. Table B.20 shows the results of comparisons between degree centrality and PageRank. Again the approaches tie for all k-values, however PageRank won in some network sizes and it is selected to be compared with *DFSH-post*. Table B.21 shows the results of comparisons between *DFSH-post* and the PageRank. The p-values are greater than 0.05 for all k-values. Therefore the approaches have a tie for all k-values.

k-value	wins	ties	losses	p-value
1%	0	21	0	1
10%	0	19	2	0.87
20%	0	19	2	0.87
30%	0	18	3	0.74
40%	0	17	4	0.62
50%	0	13	8	0.22

Table B.18: Number of wins, ties, and losses of closeness against betweenness with the p-value of binomial tests in 420 ER networks

k-value	wins	ties	losses	p-value
1%	0	21	0	1
10%	0	20	1	1
20%	0	20	1	1
30%	0	19	2	0.87
40%	0	19	2	0.87
50%	0	19	2	0.87

Table B.19: Number of wins, ties, and losses of betweenness against degree centrality with the p-value of binomial tests in 420 ER networks

k-value	wins	ties	losses	p-value
1%	0	21	0	1
10%	0	20	1	1
20%	0	20	1	1
30%	0	20	1	1
40%	0	17	4	0.62
50%	0	14	7	0.31

Table B.20: Number of wins, ties, and losses of degree centrality against PageRank with the p-value of binomial tests in 420 ER networks

k-value	wins	ties	losses	p-value
1%	2	19	0	0.87
10%	0	20	1	1
20%	0	21	0	1
30%	2	19	0	0.87
40%	3	18	0	0.74
50%	7	14	0	0.31

Table B.21: Number of wins, ties, and losses of *DFSH-post* against PageRank with the p-value of binomial tests in 420 ER networks

Table B.22 shows the results of binomial tests between closeness and betweenness approaches for ER networks, closeness wins for $k = 1\%, 10\%, 20\%, 30\%$ and betweenness wins for $k = 1\%, 40\%, 50\%$. Table B.23 shows the results between degree centrality and winners from previous table, the degree centrality wins for $k = 1\%, 30\%, 40\%$, the betweenness wins for $k = 50\%$, and cloesness wins for $k = 1\%, 10\%, 20\%$. able B.24 shows the results of comparisons between the winners from previous table and PageRank. The PageRank only ties with previous winners for $k = 1\%, 40\%$ and loses to them in other k-values. Table B.25 shows the results of comparisons between *DFSH-post* and the winners of previous table. All approaches tie for $k = 1\%$. *DFSH-post* wins for $k = 40\%, 50\%$, and centrality measures win in other k-values.

k-value	wins	ties	losses	p-value
1%	0	100	0	1
10%	78	22	0	~ 0
20%	95	5	0	~ 0
30%	95	5	0	~ 0
40%	0	19	81	~ 0
50%	0	1	99	~ 0

Table B.22: Number of wins, ties, and losses of closeness against betweenness with the p-value of binomial tests in 2000 WS networks

k-value	wins	ties	losses	p-value
1%	0	100	0	1
10%	78	22	0	~ 0
20%	69	31	0	~ 0
30%	0	1	99	~ 0
40%	0	3	97	~ 0
50%	100	0	0	~ 0

Table B.23: Number of wins, ties, and losses of closeness and betweenness against degree centrality with the p-value of binomial tests in 2000 WS networks

k-value	wins	ties	losses	p-value
1%	0	100	0	1
10%	78	22	0	~ 0
20%	95	5	0	~ 0
30%	99	1	0	~ 0
40%	0	77	23	0.097
50%	98	2	0	~ 0

Table B.24: Number of wins, ties, and losses of closeness, betweenness, and degree centrality against PageRank with the p-value of binomial tests in 2000 WS networks

k-value	wins	ties	losses	p-value
1%	0	100	0	1
10%	0	33	67	~ 0
20%	0	24	76	~ 0
30%	0	24	76	~ 0
40%	99	1	0	~ 0
50%	100	0	0	~ 0

Table B.25: Number of wins, ties, and losses of *DFSH-post* against centrality winners with the p-value of binomial tests in 2000 WS networks

Tables B.26 through B.30 show the average objective value of all approaches for some network sizes of each network model when $k = 10\%$. Tables B.31 through B.35 show the runtime of all approaches for some network sizes when $k = 10\%$.

Vertices	Close	Between	Degree	Page	*DFSH-post*
1000	14,534.4	2660.15	2304.3	2164.9	2052.5
2500	59,850.4	6609.7	5539.9	5267.25	4963.7
5000	167,578.5	12,855.3	11,052.9	10,327.7	9898.7
7500	318,831.5	19,822.3	16,820.1	15,668.7	15,151.2
10,000	390,117	26,486.3	22,726.5	21,076.2	20,699.1
12,500	598,635.9	32,886.3	28,324.6	26,255.6	26,230.9
15,000	727,728	39,531.3	33,880.7	31,588.6	31,567.4
20,000	1,226,972	52,897.1	45,441.5	42,060	42,922.5
24,000	1,529,357	63,124.85	54,470.3	50,484.5	51,920.5

Table B.26: The average objective value resulted by tested approaches for some FF network sizes when $k = 10\%$

Vertices	Close	Between	Degree	Page	*DFSH-post*
1000	8972.3	507.5	481	467.7	429.7
2500	35,737.4	1302.2	1242.1	1225.3	1108.9
5000	128,804.3	2568.8	2422.1	2379.6	2181
7500	273,865.2	3852.2	3660.1	3609.7	3285.7
10,000	442,196.1	5196.3	4815.5	4772.8	4367.3
12,500	670,012	6518.15	6060.6	6004.9	5463.5
15,000	952,523.5	7699.8	7173.6	7100	6481.1
20,000	1,230,032	10,407.3	9641.3	9564.6	8731.6
24,000	2,342,169	12,406.5	11,562.8	11,472.8	10,491.3

Table B.27: The average objective value resulted by tested approaches for some BA-m1 network sizes when $k = 10\%$

Vertices	Close	Between	Degree	Page	*DFSH-post*
1000	251,383.6	10,108.3	6208.2	4783.8	4996.5
2500	1,873,572	94,825.3	20,460.1	17,568.3	16,136.9
5000	8,012,312	550,725.8	44,078.6	30,879.3	30,985.4
7500	18,523,847	1,646,337	57,445.7	51,393.9	42,928.9
10,000	33,875,394	3,834,258	82,572.1	61,933.1	61,740.7
12,500	52,875,120	6,923,655	90278.2	73,422	71,445.5
15,000	76,876,687	10,464,912	103,972.6	84,152.1	84,367.2
20,000	137,918,358.6	23,084,817	187,129.8	136,747.6	130,837.1
24,000	199,910,299.3	37,216,932	245,117.1	179,745.3	178,688.8

Table B.28: The average objective value resulted by tested approaches for some BA-m2 network sizes when $k = 10\%$

Vertices	Close	Between	Degree	Page	*DFSH-post*
1000	404,505.1	404,550	404,550	404,550	404,550
2500	2,529,450	2,530,125	2,530,125	2,530,125	2,529,900
5000	10,116,003	10,122,750	10,120,728	10,122,750	10,121,400
7500	22,763,033	22,774,839	22,775,176	22,777,875	22,774,838
10,000	40,480,205	40,495,050	40,494,150	40,495,050	40,492,350
12,500	63,232,890	63,273,376	63,264,381	63,275,625	63,270,000
15,000	91,062,247	91,116,225	91,112,851	91,118,250	91,109,475
20,000	161,887,531	161,990,100	161,965,809	161,991,000	161,976,601
24,000	233,112,651	233,268,120	233,220,615	233,269,200	233,248,681

Table B.29: The average objective value resulted by tested approaches for some WS network sizes when $k = 10\%$

Vertices	Close	Between	Degree	Page	*DFSH-post*
1000	404,550	404,550	404,550	404,550	404,550
2000	1,619,100	1,619,100	1,619,100	1,619,100	1,619,100
3000	3,643,650	3,643,515	3,643,650	3,643,515	3,643,650
4000	6,478,200	6,478,200	6,478,200	6,478,200	6,478,200
5000	10,122,750	10,122,750	10,122,750	10,122,750	10,122,750

Table B.30: The average objective value resulted by tested approaches for some ER network sizes when $k = 10\%$

Vertices	Close	Between	Degree	Page	*DFSH-post*
1000	0.25	0.32	0.007	0.02	0.031
2500	1.79	2.12	0.01	0.05	0.09
5000	3.41	5.26	0.01	0.23	0.51
7500	6.78	7.53	0.03	0.27	0.75
10,000	9.15	13.85	0.04	0.31	0.82
12,500	14.82	22.94	0.04	0.35	0.89
15,000	21.35	33.18	0.06	0.45	0.99
20,000	38.75	64.11	0.06	0.76	1.24
24,000	56.04	164.44	0.11	1.24	1.86

Table B.31: The average runtime of approaches for some FF network sizes when $k = 10\%$

Vertices	Close	Between	Degree	Page	*DFSH-post*
1000	0.06	0.102	0.005	0.016	0.025
2500	0.45	0.63	0.01	0.024	0.046
5000	1.73	3.007	0.016	0.061	0.078
7500	4.07	6.36	0.019	0.073	0.124
10,000	6.89	11.87	0.028	0.104	0.1872
12,500	11.37	20.56	0.043	0.136	0.265
15,000	19.58	30.74	0.059	0.18	0.436
20,000	28.305	62.133	0.088	0.231	0.577
24,000	46.75	100.74	0.13	0.285	0.697

Table B.32: The average runtime of approaches for some BA-m1 network sizes when $k = 10\%$

Vertices	Close	Between	Degree	Page	*DFSH-post*
1000	0.083	0.142	0.004	0.0168	0.022
2500	0.58	1.08	0.008	0.025	0.076
5000	2.71	4.3	0.015	0.064	0.265
7500	6.64	9.23	0.022	0.0737	0.545
10,000	10.05	16.7	0.049	0.105	0.983
12,500	14.49	28.23	0.051	0.136	1.611
15,000	29.71	41.55	0.054	0.19	2.361
20,000	37.77	82.33	0.1	0.24	4.24
24,000	56.5	125.48	0.113	0.291	5.637

Table B.33: The average runtime of approaches for some BA-m2 network sizes when $k = 10\%$

Vertices	Close	Between	Degree	Page	*DFSH-post*
1000	0.137	0.202	0.001	0.028	0.044
2500	0.767	1.21	0.033	0.057	0.113
5000	3.15	6.08	0.059	0.089	0.361
7500	9.02	11.6	0.097	0.17	0.808
10,000	12.26	21.13	0.13	0.295	1.461
12,500	20.29	35.69	0.16	0.33	2.199
15,000	28.47	55.27	0.23	0.59	3.232
20,000	57.83	111.8	0.6	1.24	5.868
24,000	94.86	171.28	0.98	2.04	9.038

Table B.34: The average runtime of approaches for some WS network sizes when $k = 10\%$

Vertices	Close	Between	Degree	Page	*DFSH-post*
1000	0.216	0.461	0.008	0.08	0.082
2000	0.84	1.465	0.015	0.17	0.251
3000	1.58	3.299	0.023	0.3	0.451
4000	7.68	8.29	0.051	0.49	1.032
5000	33.87	11.75	0.083	0.76	1.583

Table B.35: The average runtime of approaches for some ER network sizes when $k = 10\%$

Appendix C

Extra tables for real-world networks

Tables C.1 through C.14 show the objective values resulted by *DFSH-post* and proposed centrality measures for each tested real-world network. The lowest objective value for each k-value is bolded in all tables representing the objective values resulted by tested approaches on real-world networks. Moreover, the centrality measures with time complexity more than 4 hours are starred in the following tables since their resulting objective values were not calculated. Table C.15 represents the average runtime of each approach for tested real-world networks.

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	25,651	25,651	**23,018**	25,651	25,651
10%	**7771**	12,571	8553	12,729	8202
20%	2425	1650	2104	1048	**784**
30%	1109	277	233	214	**200**
40%	**54**	138	54	63	72
50%	472	97	22	43	**20**

Table C.1: The objective values resulted by proposed methodology and centrality measures for the USAir97 network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	104,415	76,332	**57,604**	118,382	101,695
10%	**1551**	7301	2380	2810	1602
20%	**415**	3463	605	767	457
30%	**202**	3067	245	321	229
40%	**97**	2204	164	188	139
50%	**69**	1491	110	91	79

Table C.2: The objective values resulted by proposed methodology and centrality measures for the Human-disease network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	372,077	345,580	**264,313**	397,387	346,798
10%	**8603**	156,616	17,259	267,929	12,152
20%	**3395**	29,277	4571	47,236	3715
30%	2363	16,907	2563	8679	**2327**
40%	1533	7154	1905	3173	**1496**
50%	4747	5561	1486	1402	**852**

Table C.3: The objective values resulted by proposed methodology and centrality measures for the Gene-disease network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	714,031	1,164,020	875,417	**697,455**	**697,455**
10%	3013	335,987	5500	**2199**	2855
20%	**247**	157,614	1132	382	328
30%	**85**	63,687	318	132	88
40%	11	9122	50	24	**4**
50%	**0**	5901	**0**	**0**	**0**

Table C.4: The objective values resulted by proposed methodology and centrality measures for the Bipartite-disease network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	288,636,371	288,684,427	**287,484,254**	288,684,426	287,964,024
10%	232,643,278	234,826,996	**219,440,950**	232,859,030	228,840,988
20%	174,704,882	179,106,274	**151,981,162**	176,391,235	169,620,727
30%	117,175,433	129,178,886	**100,203,918**	123,551,535	114,632,871
40%	**45,819,811**	88,531,739	57,315,865	76,577,025	62,145,722
50%	20,723,034	54,962,777	21,021,169	39,681,976	**13,798,820**

Table C.5: The objective values resulted by proposed methodology and centrality measures for the Hep-citation network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	256,857,903	347,632,648	**206,069,905**	237,191,707	210,764,154
10%	38,502,278	66,199,754	39,916,753	**12,194,554**	16,732,757
20%	20,369,891	15,806,464	64,831	23,811	**15,556**
30%	**5776**	8,388,754	10,445	9263	5953
40%	**2007**	1,950,125	4673	4686	2620
50%	**142**	300,013	4202	2316	962

Table C.6: The objective values resulted by proposed methodology and centrality measures for the Email network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	1,616,046,583	1,735,459,194	**1,539,264,942**	1,605,262,847	1,569,876,622
10%	775,254,833	803,464,151	**580,195,651**	665,268,053	628,724,144
20%	424,672,334	291,697,307	142,731,054	141,083,564	**115,109,115**
30%	59,621,496	68,292,199	159,248	45,462	**11,393**
40%	15,514	11,629,705	**866**	3613	1259
50%	**31**	1,213,488	51	900	60

Table C.7: The objective values resulted by proposed methodology and centrality measures for the Marker network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	4,306,899,559	6,685,356,322	5,166,835,906	4,113,548,173	**3,556,134,525**
10%	44,648,398	3,689,516,359	5,478,896	**525,057**	11,336,152
20%	1,468,811	524,445,407	196,927	91,862	**70,756**
30%	**21,271**	81,942,949	64,010	43,529	27,916
40%	**10,323**	13,246,588	34,547	21,122	12,055
50%	**699**	2,138,414	10,387	11,795	3275

Table C.8: The objective values resulted by proposed methodology and centrality measures for the Internet network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	283,868,160,293	*	*	253,686,945,273	**230,329,625,785**
10%	25,745,386,142	*	*	3,019,233	**1,295,641**
20%	200,688	*	*	273,677	**123,630**
30%	114,004	*	*	77,534	**23,328**
40%	67,621	*	*	30,261	**1734**
50%	**0**	*	*	**0**	191

Table C.9: The objective values resulted by proposed methodology and centrality measures for the Youtube network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	575,151,203,833	*	*	571,890,030,386	**556,704,021,424**
10%	344,989,315,347	*	*	**230,105,957,947**	292,894,795,037
20%	164,919,474,670	*	*	**5,849,179,203**	9,398,995,959
30%	1,053,346,609	*	*	**15,058,025**	517,438,486
40%	**1,844,663**	*	*	11,063,077	8,982,761
50%	**477,985**	*	*	8,196,982	1,448,127

Table C.10: The objective values resulted by proposed methodology and centrality measures for the Pennsylvania-road network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	862,213,811,375	*	*	858,378,888,866	**853,725,302,901**
10%	493,706,782,203	*	*	**112,916,600,319**	408,318,794,530
20%	231,972,271,948	*	*	**1,838,794,178**	11,803,235,140
30%	101,478,310	*	*	**51,523,517**	357,097,492
40%	**3,287,369**	*	*	36,119,420	19,741,490
50%	**495,692**	*	*	26,099,008	2,503,038

Table C.11: The objective values resulted by proposed methodology and centrality measures for the Texas-road network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	1,866,249,228,277	*	*	1,846,912,628,856	**1,790,402,565,748**
10%	799,756,015,785	*	*	**726,479,849,759**	906,628,514,505
20%	137,533,268,725	*	*	**11,716,704,553**	61,164,574,748
30%	1,984,295,146	*	*	**49,193,827**	917,070,521
40%	**4,518,111**	*	*	38,674,640	18,129,542
50%	**997,841**	*	*	22,929,879	2,858,888

Table C.12: The objective values resulted by proposed methodology and centrality measures for the California-road network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	1,172,220,746,606	*	*	1,115,012,962,937	**1,003,797,127,054**
10%	132,576,427,963	*	*	129,907,089,624	**31,382,904,785**
20%	13,703,056,599	*	*	48,767,651	**38,801,672**
30%	**566,457**	*	*	2,522,271	2,130,481
40%	**234,381**	*	*	754,389	273,257
50%	**46,539**	*	*	337,659	97,333

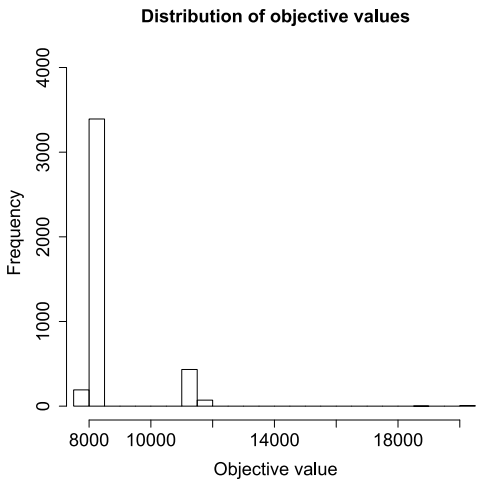
Table C.13: The objective values resulted by proposed methodology and centrality measures for the Skitter network

k-value	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
1%	7,557,955,150,804	*	*	7,552,454,742,641	**7,443,750,704,418**
10%	5,031,050,710,705	*	*	5,014,236,886,729	**4,688,857,693,021**
20%	2,545,370,022,349	*	*	2,606,582,691,075	**2,174,729,931,569**
30%	1,038,044,417,245	*	*	550,914,138,833	**183,943,902,653**
40%	266,142,346,439	*	*	**5,049,598**	586,570,364
50%	**89,782**	*	*	817,307	322,298

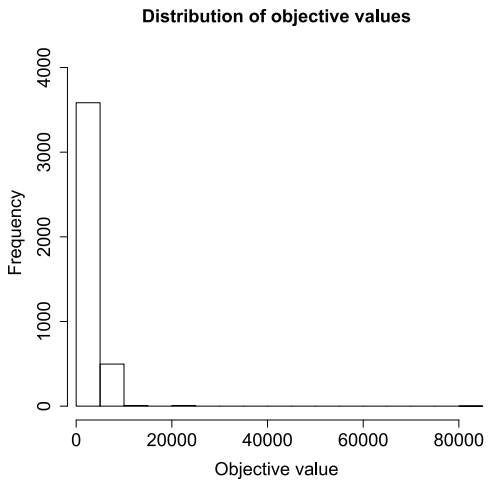
Table C.14: The objective values resulted by proposed methodology and centrality measures for the Live-journal network

Network	*DFSH-post*	Closeness	Betweenness	Degree	PageRank
USAir97	0	0	0.015	0	0
Human Disease	0.001	0.015	0.031	0	0.001
Gene Disease	0.01	0.062	0.078	0.001	0.015
Bipartite Disease	0.003	0.109	0.171	0	0.015
Hep-citation	2.184	105.86	225.108	0.025	0.372
Email	1.411	152.533	278.435	0.015	0.408
Marker	31.668	1550.187	3653.791	0.046	0.967
Internet	6.318	1873.666	4143.463	0.031	0.823
Youtube	808.2	*	*	0.702	19.28
PA Road	90.14	*	*	0.68	12.63
TX Road	170.08	*	*	0.74	15.444
CA Road	309.16	*	*	0.93	22.62
Skitter	4559.591	*	*	1.34	27.580
Live Journal	138,810.677	*	*	2.591	74.49

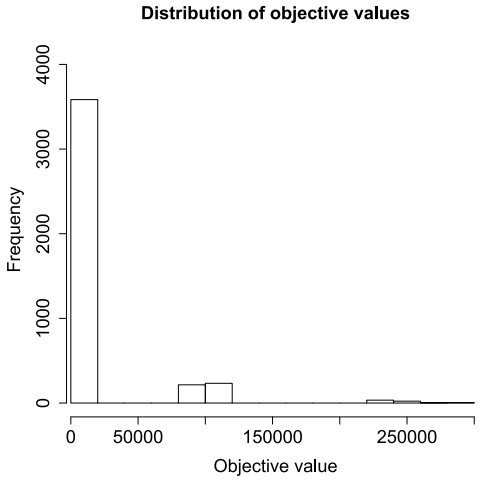
Table C.15: The runtime of each approach in seconds for the 14 tested real-world networks



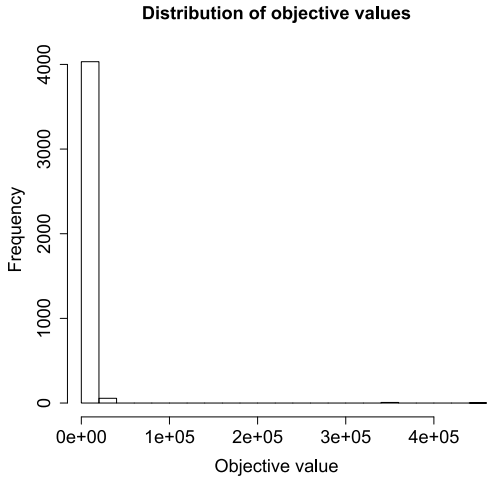
(a) USAir97 network



(b) Human Disease network

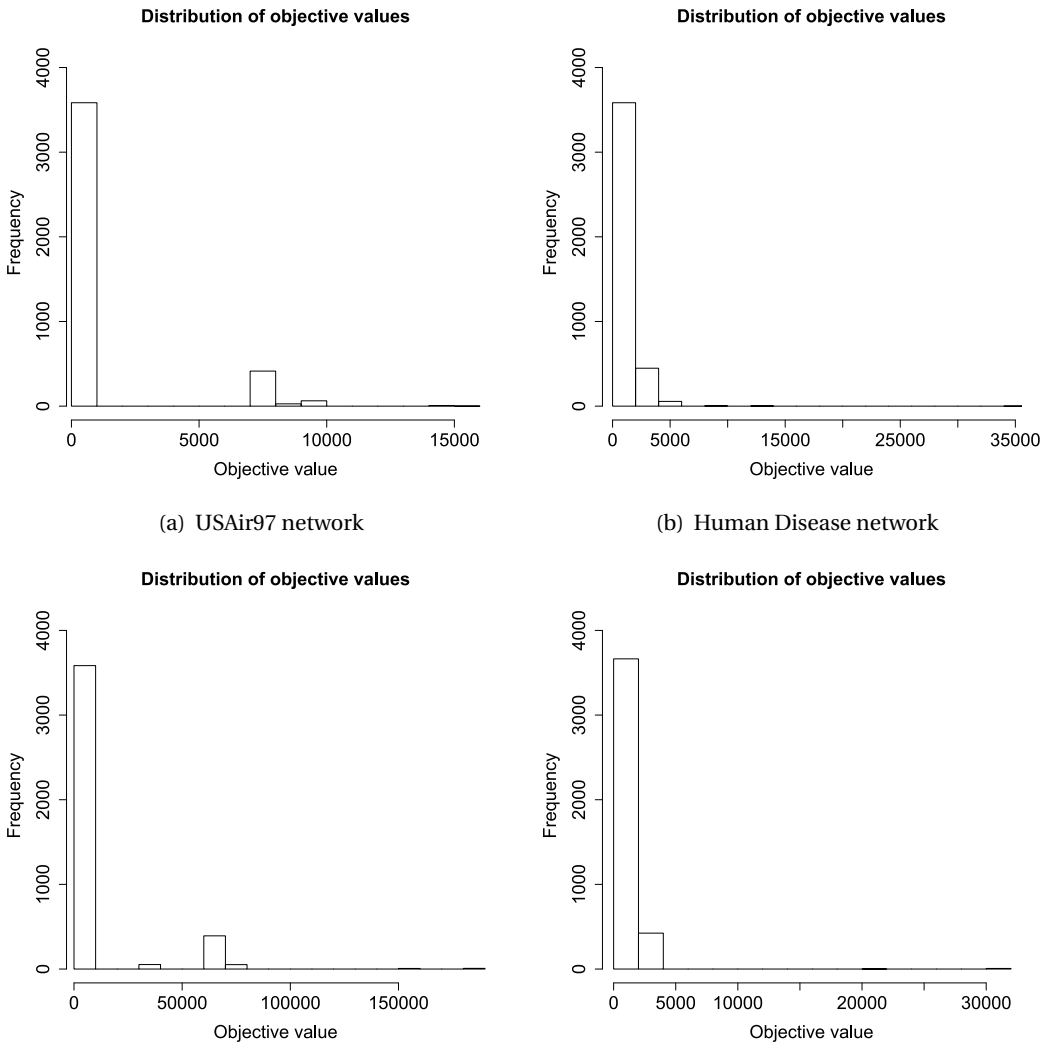


(c) Gene Disease network



(d) Bipartite Disease network

Figure C.1: The distribution of objective values across all weights are plotted for the 4 small real-world networks USAir97, Human Disease, Gene Disease, and Bipartite Disease networks when $k = 10\%$



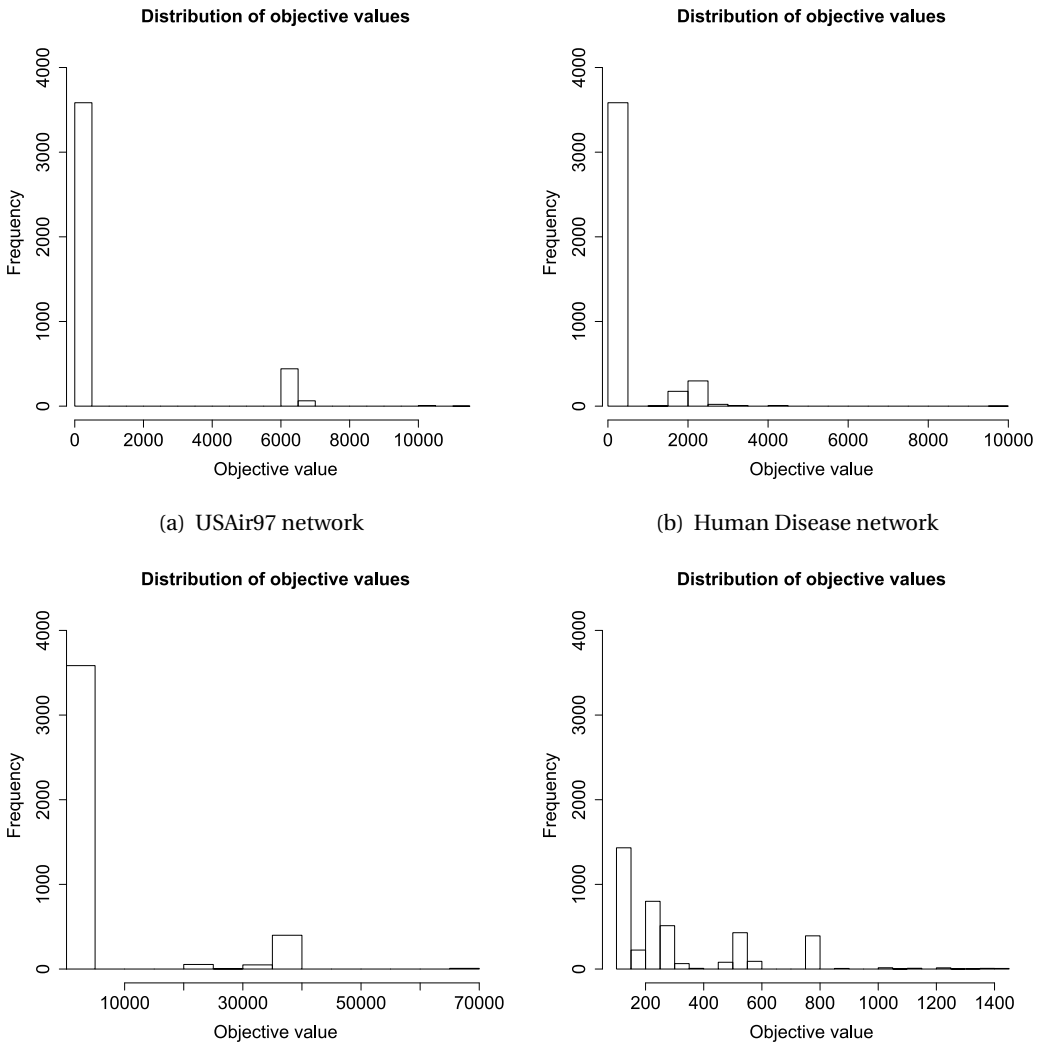
(a) USAir97 network

(b) Human Disease network

(c) Gene Disease network

(d) Bipartite Disease network

Figure C.2: The distribution of objective values across all weights are plotted for the 4 small real-world networks USAir97, Human Disease, Gene Disease, and Bipartite Disease networks when $k = 20\%$



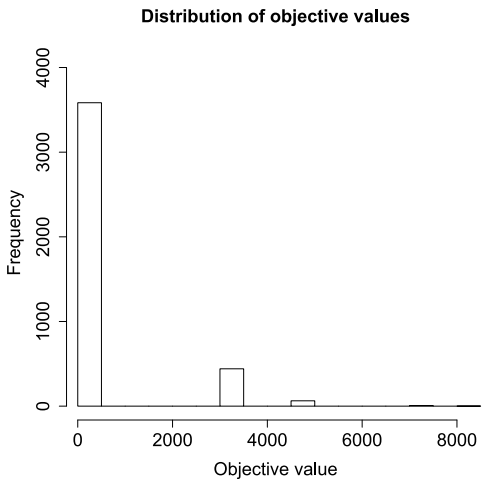
(a) USAir97 network

(b) Human Disease network

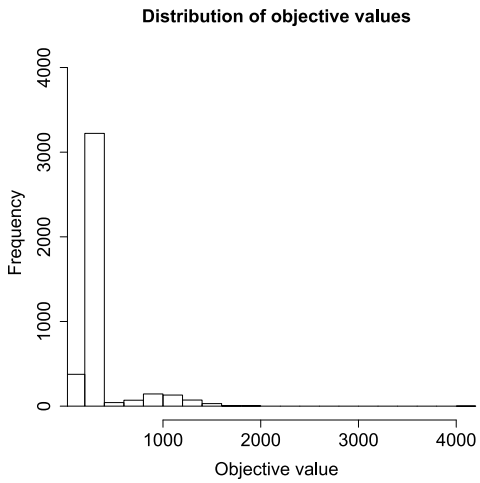
(c) Gene Disease network

(d) Bipartite Disease network

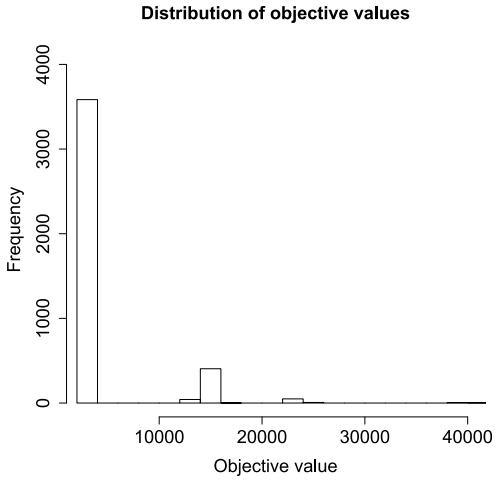
Figure C.3: The distribution of objective values across all weights are plotted for the 4 small real-world networks USAir97, Human Disease, Gene Disease, and Bipartite Disease networks when $k = 30\%$



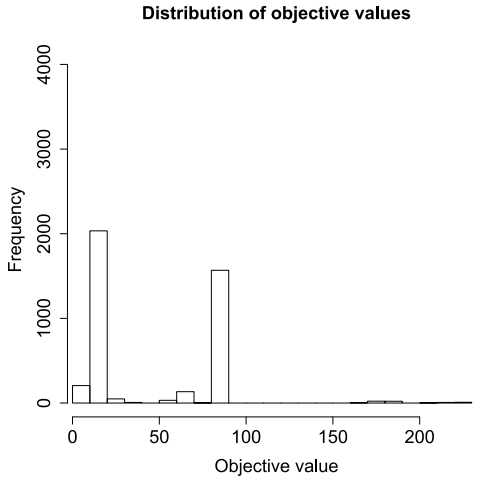
(a) USAir97 network



(b) Human Disease network

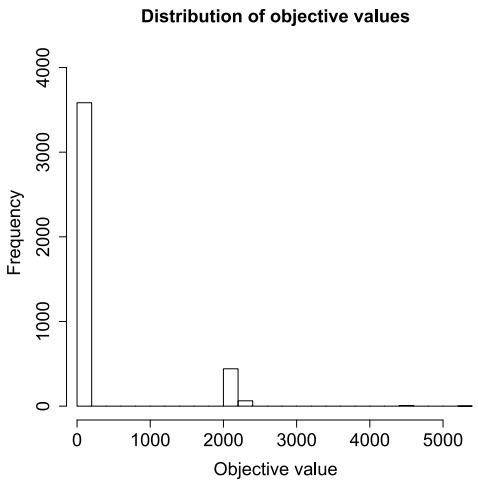


(c) Gene Disease network

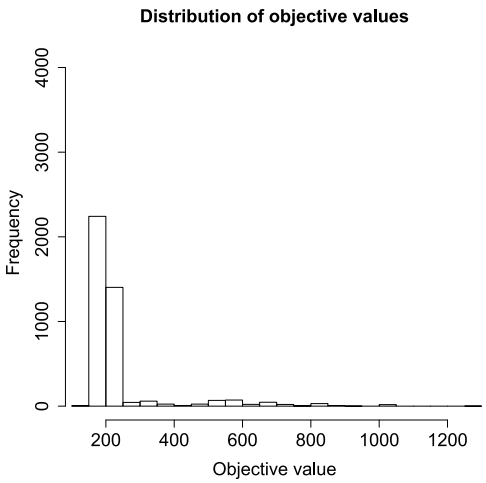


(d) Bipartite Disease network

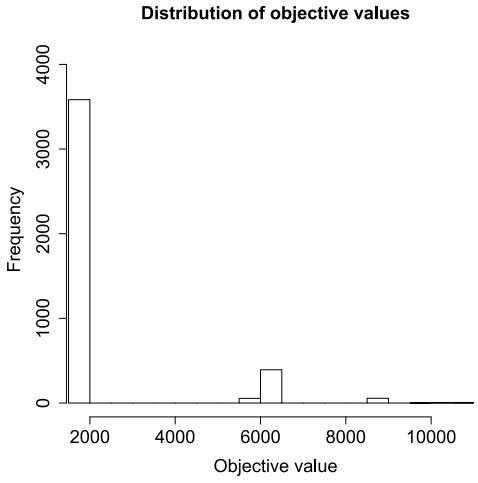
Figure C.4: The distribution of objective values across all weights are plotted for the 4 small real-world networks USAir97, Human Disease, Gene Disease, and Bipartite Disease networks when $k = 40\%$



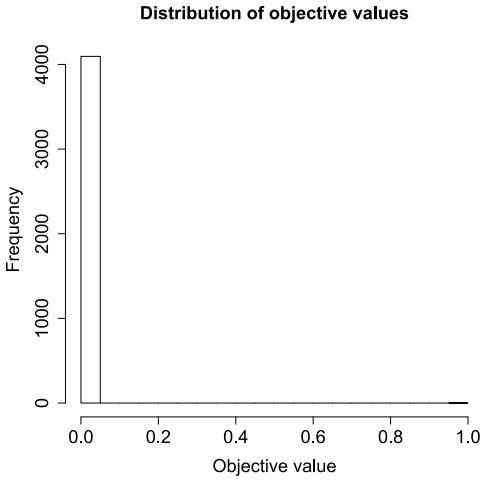
(a) USAir97 network



(b) Human Disease network



(c) Gene Disease network



(d) Bipartite Disease network

Figure C.5: The distribution of objective values across all weights are plotted for the 4 small real-world networks USAir97, Human Disease, Gene Disease, and Bipartite Disease networks when $k = 50\%$